

**PENERAPAN ANALISIS SENTIMEN UNTUK MENILAI SUATU
PRODUK PADA TWITTER BERBAHASA INDONESIA DENGAN
METODE NAÏVE BAYES CLASSIFIER DAN INFORMATION
GAIN**

SKRIPSI

Untuk memenuhi sebagian persyaratan
memperoleh gelar Sarjana Komputer

Disusun oleh:
AHMAD WILDAN ATTABI'
NIM:135150201111185



PROGRAM STUDI TEKNIK INFORMATIKA
JURUSAN TEKNIK INFORMATIKA
FAKULTAS ILMU KOMPUTER
UNIVERSITAS BRAWIJAYA
MALANG
2018

PENGESAHAN

PENERAPAN ANALISIS SENTIMEN UNTUK MENILAI SUATU PRODUK PADA
TWITTER BERBAHASA INDONESIA DENGAN METODE NAÏVE BAYES CLASSIFIER
DAN INFORMATION GAIN

SKRIPSI

Diajukan untuk memenuhi sebagian persyaratan
memperoleh gelar Sarjana Komputer

Disusun Oleh :
Ahmad Wildan Attabi'
NIM: 135150201111185

Skripsi ini telah diuji dan dinyatakan lulus pada
23 Maret 2018

Telah diperiksa dan disetujui oleh:

Dosen Pembimbing I



Lailil Muflikhah, S.Kom, M.Sc
NIP: 197411132005012001

Dosen Pembimbing II



M. Ali Fauzi. S.Kom, M.Kom
NIK: 2015028901011001

Mengetahui

Ketua Jurusan Teknik Informatika




Tri Astoto Kurniawan, S.T, M.T, Ph.D
NIP: 197105182003121001

PERNYATAAN ORISINALITAS

Saya menyatakan dengan sebenar-benarnya bahwa sepanjang pengetahuan saya, di dalam naskah skripsi ini tidak terdapat karya ilmiah yang pernah diajukan oleh orang lain untuk memperoleh gelar akademik di suatu perguruan tinggi, dan tidak terdapat karya atau pendapat yang pernah ditulis atau diterbitkan oleh orang lain, kecuali yang secara tertulis disitasi dalam naskah ini dan disebutkan dalam daftar pustaka.

Apabila ternyata didalam naskah skripsi ini dapat dibuktikan terdapat unsur-unsur plagiasi, saya bersedia skripsi ini digugurkan dan gelar akademik yang telah saya peroleh (sarjana) dibatalkan, serta diproses sesuai dengan peraturan perundang-undangan yang berlaku (UU No. 20 Tahun 2003, Pasal 25 ayat 2 dan Pasal 70).

Malang, 23 Maret 2018



Ahmad Wildan Attabi'

NIM: 135150201111185

KATA PENGANTAR

Puji syukur atas kehadiran Tuhan Yang Maha Esa yang selalu memberikan Anugerah dan KeberkahanNya kepada Penulis sehingga laporan skripsi yang berjudul “Penerapan Analisis Sentimen untuk Menilai Suatu Produk pada Twitter Berbahasa Indonesia dengan Metode Naive Bayes dan Informarion Gain” ini dapat diselesaikan dengan baik. Skripsi ini disusun untuk memenuhi syarat memperoleh gelar Sarjana Komputer.

Penulis menyadari pula bahwa selama penyusunan skripsi ini tidak akan berhasil tanpa adanya bantuan dan dukungan dari banyak pihak. Oleh sebab itu, penulis ingin menyampaikan rasa terima kasih dan penghormatan yang sebesar-besarnya kepada:

1. Ibu Lailil Muflikhah, S. Kom, M.Sc dan Bapak M. Ali Fauzi, S.Kom, M.Kom selaku dosen pembimbing skripsi yang telah membimbing dan mengarahkan penulis sehingga dapat menyelesaikan skripsi dengan baik.
2. Bapak Wayan Firdaus Mahmudy, S.Si, M.T, Ph.D, Bapak Ir. Heru Nurwarsito, M.Kom, Bapak Drs. Marji, M.T, dan Bapak Edy Santoso, S.Si, M.Kom selaku Dekan, Wakil Dekan 1, Wakil Dekan 2, dan Wakil Dekan 3 Fakultas Ilmu Komputer Universitas Brawijaya Malang.
3. Seluruh Dosen dan Staff Teknik Informatika Fakultas Ilmu Komputer Universitas Brawijaya yang telah banyak memberi bantuan dan dukungan selama penulis menempuh studi di Universitas Brawijaya dan selama pengerjaan skripsi ini.
4. Pihak Twitter sebagai objek untuk penelitian skripsi ini hingga laporan skripsi ini selesai.
5. Bapak Besar Handoko, SE. dan Ibu Masnaimah yang selalu mendoakan, memberikan kasih sayang dan memotivasi penulis untuk tetap semangat dalam menyelesaikan skripsi, sekaligus menjadi orang tua kandung yang luar biasa selama ini, serta seluruh keluarga besar atas segala nasehat, perhatian, dan dorongannya menambahkan kekuatan jiwa dan semangat yang selalu terjaga bagi penulis.
6. Gilang Pratama, dan Dimas Joko Haryanto, dan M. Basri, Spd. yang membantu memberikan nasehat, bantuan, dan pengarahan secara langsung guna terselesaikan skripsi ini.
7. Krisna Hayuning Fadlilah, Psi. yang memberikan semangat, bantuan, dan kebersamaan, serta secara tidak langsung mengajarkan arti makna bersabar dan ikhlas, sehingga menjadi seseorang yang berarti bagi penulis.

8. Rinaldi Oktavianto Airlangga, Endra Kurniawan, Nur Fahmi Amirudin, Riken Afandi, Winarsih, Novi Fitriani, Bayu Rizka Eka Mahardika, Moch. Hilman Khoeruddin, dan M. Hilmi Kurniawan, yang telah menjadi orang-orang yang selalu baik dan terus memberikan semangat, dorongan, kebersamaan, dan pengalaman sosial yang tak terlupakan bagi penulis.
9. Sahabat – Sahabati Musa Al Jabar yang secara tidak langsung memberikan dukungan dan memfasilitasi penulis untuk mempermudah pengerjaan skripsi ini
10. Sahabat Kontrakan dan Sahabat Komunitas GPAN Malang yang tiada hentinya memberikan semangat dan dukungan, serta kebersamaan yang sangat berarti bagi Penulis.
11. Dan seluruh teman dan orang-orang baik yang tidak mungkin saya sebutkan satu-satu yang selalu memberikan dampak positif bagi penulis.

Malang, 23 Maret 2018

Ahmad Wildan Attabi'
whildany@gmail.com



ABSTRAK

Twitter memiliki peran besar dalam mempengaruhi aspek dari segi sosial, komunikasi, psikologi, marketing, dan politik. Kiriman Tweet komentar atau opini secara tidak langsung akan menjadi sebuah review terhadap penilaian pada suatu produk tersebut. Salah satu sektor produk yang sangat diminati oleh banyak orang saat ini adalah produk kecantikan dan perawatan kulit. Mereka mencari produk yang memiliki kualitas terbaik dengan cara melihat pendapat dari orang lain, sehingga mereka memiliki gambaran yang mempengaruhi minat mereka atas pendapat orang lain yang disampaikan lewat Twitter terkait hasilnya setelah memakai produk tersebut. Analisis sentimen dapat membantu dalam menganalisis dan mengklasifikasikan menjadi kelas positif dan kelas negatif dari pendapat orang yang ada di twitter terkait produk yang bersangkutan untuk dicari nilai kecenderungan pasarnya dan kualitas produk pada pandangan publik. Opini dan komentar terkait produk Mustika Ratu dipilih untuk mewakili obyek penelitian ini, dengan alasan pertumbuhan ekonomi dan banyaknya pengguna dari Mustika Ratu yang merupakan perusahaan di bidang perawatan kulit dan kecantikan. Metode *Naïve Bayes Classifier* dipilih untuk alasan kemudahan dalam implementasinya, dan memiliki performa cepat dalam melatih, sedangkan penambahan *Information Gain* diperlukan untuk proses seleksi fitur dengan mengurangi keberadaan kata yang tidak relevan pada data yang digunakan. Pengujian dilakukan dengan jumlah data sebanyak 200 (100 dokumen positif, dan 100 dokumen negatif) menggunakan kisaran nilai threshold : 0, 0.01, 0.02, 0.03, 0.04, 0.05, 0.06, 0.07, 0.08, 0.09, dan 0.10. Hasil yang diperoleh terdapat peningkatan akurasi dengan selisih 4%, nilai rata-rata akurasi jika tanpa menggunakan metode *Information Gain* (threshold 0) sebesar 70%, sedangkan dengan menggunakan metode *Information Gain* (threshold 0.01) sebesar 74%. Hal ini dipengaruhi oleh beberapa faktor diantaranya jumlah data latih dan persebaran jenis kata pada data latih dokumen positif maupun dokumen negatif. Nilai akurasi yang paling tinggi didapatkan pada K1 (threshold 0.02), kemudian K5, K6 (threshold 0.01), dan K7 (threshold 0.02 dan 0.08) dengan presentase 85%, sedangkan pada k bagian tertentu dengan threshold tertentu memperoleh titik terendah sebesar 50%.

Kata Kunci : Analisis Sentimen, *Naive Bayes*, *Information Gain*, Twitter, Mustika Ratu.

ABSTRACT

Twitter has a major role in the development of social, communication, psychological, marketing and political aspects. Posts Tweet comments or review indirectly will be a review of the assessment on a product. One of the most sought after products sectors today is beauty and skin care products. They look for products that they share with others, so they have a picture that affects their interest on the opinions of others who delivered via Twitter related results after using the product. Sentiment analysis can help in analyzing and classifying into positive and negative terms of twitter-related opinions about product trends and product quality in the public view. Opinions and comments related to Mustika Ratu's products are the subject of this study, citing the economic growth and the large number of users of Musitka Ratu who are companies in the field of beauty and beauty care. The Naïve Bayes Classifier method is selected for implementation use, and has a fast performance in training, while the addition of Information is required for the feature selection process by reducing the presence of irrelevant words in the data used. The test is performed with 200 data (100 positive documents, and 100 negative documents) using the thresholds : 0, 0.01, 0.02, 0.03, 0.04, 0.05, 0.06, 0.07, 0.08, 0.09, dan 0.10. The results obtained are adjusted for a difference of 4%, the highest average value if no Information Gain (threshold 0) is 70%, while using Information Gain (threshold 0.01) equal to 74%. This is influenced by several factors such as the amount of data and data that spread from data data and documents. The highest accuracy value is obtained at K1 (threshold 0,02), then K5, K6 (threshold 0.01), and K7 (threshold 0,02 and 0,08) with percentage 85%, while at k with threshold at the lowest point 50%.

Keywords: Analysis Sentiment, Naive Bayes, Information Gain, Twitter, Mustika Ratu.

DAFTAR ISI

COVER.....	i
PENGESAHAN	ii
PERNYATAAN ORISINALITAS	iii
KATA PENGANTAR	iv
ABSTRAK	vi
ABSTRACT	vii
DAFTAR ISI.....	viii
DAFTAR TABEL.....	xi
DAFTAR GAMBAR.....	xii
DAFTAR KODE PROGRAM.....	xiii
DAFTAR LAMPIRAN.....	xiv
BAB 1 PENDAHULUAN.....	1
1.1 Latar belakang.....	1
1.2 Rumusan masalah.....	4
1.3 Tujuan	4
1.4 Manfaat.....	5
1.5 Batasan masalah	5
1.6 Sistematika Pembahasan	5
BAB 2 LANDASAN KEPUSTAKAAN	7
2.1 Kajian Pustaka	7
2.2 Text Mining	10
2.2.1 Text Preprocessing	10
2.2.2 Feature Selection	11
2.3 Analisis Sentimen	12
2.3.1 Opinion Lexicon.....	12
2.4 Information Gain.....	13
2.5 TF Biner	14
2.6 Naive Bayes Classifier	15
2.7 Twitter.....	16
2.8 Penilaian Produk	16

2.9 Framework	167
2.9.1 MVC	17
2.9.2 PHP	17
BAB 3 METODOLOGI	18
3.1 Studi Literatur	18
3.2 Pengumpulan Data	19
3.3 Analisis Kebutuhan	19
3.3.1 Kebutuhan Fungsional.....	19
3.3.2 Kebutuhan Non Fungsional.....	19
3.4 Perancangan Sistem.....	20
BAB 4 ANALISIS DAN PEMBAHASAN	21
4.1 Deskripsi Umum Sistem	21
4.2 Perancangan Sistem.....	21
4.2.1Preprocessing.....	22
4.2.1.1 Case Folding	22
4.2.1.2 Tokenizing	22
4.2.1.3 Normalisasi Bahasa	23
4.2.1.4 Filtering	23
4.2.1.5 Steaming	24
4.2.2 TF-Biner.....	24
4.2.3 Proses Information Gain	25
4.2.4 Proses Naive Bayes Classifier.....	25
4.3 Manualisasi Perhitungan	26
4.3.1 Dataset.....	26
4.3.2 Menghitung TF Biner	30
4.3.3 Manualisasi Information Gain	31
4.3.4 Klasifikasi Dokumen dengan Naive Bayes Classifier	34
4.4 Perancangan Pengujian	39
BAB 5 IMPLEMENTASI	40
5.1 Implementasi Program	40
5.1.1 Load Dokumen.....	44
5.1.2 Implementasi Preprocessing	49

5.1.3 Implementasi TF-Biner	58
5.1.4 Implementasi Information Gain	60
5.1.5 Implementasi Naive Bayes	69
5.1.6 Implementasi Testing	81
5.1.7 Implementasi Testing Data untuk 10 Fold Cross Validation	87
5.2 Implementasi Antarmuka Sistem	96
5.2.1 Interface Load Dokumen	96
5.2.2 Interface Preprocessing.....	97
5.2.3 Interface Klasifikasi.....	99
5.2.4 Interface Testing.....	100
BAB 6 PENGUJIAN DAN ANALISIS.....	101
6.1 Pengujian	101
6.1.1 Pengujian dan analisis validasi terhadap data dengan metode 10-fold cross validation.....	101
BAB 7 KESIMPULAN	1018
7.1 Kesimpulan.....	109
7.2 Saran	110
DAFTAR PUSTAKA.....	111

DAFTAR TABEL

Tabel 2.1 Tabel Kajian Pustaka.....	8
Tabel 2.2 Tabel Stopword List.....	11
Tabel 4.1 Dataset Perhitungan Manualisasi.....	27
Tabel 4.2 Hasil Tokenisasi	28
Tabel 4.3 Hasil Normalisasi Bahasa.....	29
Tabel 4.4 Hasil Akhir <i>Preprocessing</i>	30
Tabel 4.5 Manualisasi Perhitungan.....	31
Tabel 4.6 Perhitungan Entropy Total.....	32
Tabel 4.7 Perhitungan Entropy fitur “biar” setelah pemisahaan.....	32
Tabel 4.8 Perhitungan <i>Information Gain</i> Fitur “biar”	33
Tabel 4.9 Hasil Keseluruhan Nilai InfoGain(A).....	33
Tabel 4.10 Langkah Perhitungan Prior.....	34
Tabel 4.11 Langkah Perhitungan Likelihood kata “biar”	34
Tabel 4.12 Hasil Keseluruhan Perhitungan Likelihood tiap Fitur/Kata.....	35
Tabel 4.13 Hasil Klasifikasi <i>Naive Bayes</i> terhadap Dokumen 1.....	35
Tabel 4.14 Hasil Klasifikasi <i>Naive Bayes</i> terhadap Dokumen 2.....	36
Tabel 4.15 Hasil Klasifikasi <i>Naive Bayes</i> terhadap Dokumen 3.....	36
Tabel 4.16 Hasil Klasifikasi <i>Naive Bayes</i> terhadap Dokumen 4.....	37
Tabel 4.17 Hasil Klasifikasi <i>Naive Bayes</i> terhadap Dokumen 5.....	37
Tabel 4.18 Hasil Klasifikasi <i>Naive Bayes</i> terhadap Dokumen 6.....	38
Tabel 4.19 <i>Matrix Confusion</i> untuk mencari Nilai Akurasi	39
Tabel 5.1 Daftar Fungsi	40
Tabel 6.1 Tabel Akurasi Tiap K bagian.....	101
Tabel 6.2 – Hasil klasifikasi dengan data latih 60 data.....	103
Tabel 6.3 – Hasil Klasifikasi dengan data latih 120 data.....	103
Tabel 6.4 – Hasil Klasifikasi K3 dengan threshold 0.01.....	104
Tabel 6.5 – Jenis kata dan jumlah kata pada dokumen 6 K3.....	105
Tabel 6.6 – Tabel nilai likelihood saat threshold Information Gain bernilai 1	106
Tabel 6.7 – Hasil Klasifikasi K3 dengan threshold 0.10.....	106
Tabel 6.8 – Hasil Klasifikasi K3 dengan data latih dan data uji sama.....	107

DAFTAR GAMBAR

Gambar 2.1 Tahapan proses text preprocessing	10
Gambar 3.1 Tahapan Penelitian.....	18
Gambar 4.1 Diagram Alir Sistem.....	21
Gambar 4.2 Diagram Alir <i>Preprocessing</i>	22
Gambar 4.3 Diagram Alir Normalisasai Bahasa.....	23
Gambar 4.4 Diagram Alir TF-Biner.....	24
Gambar 4.5 Diagram Alir <i>Information Gain</i>	25
Gambar 4.6 Diagram Alir <i>Naive Bayes</i>	26
Gambar 5.1 Load Dokumen (Tampilan Awal).....	96
Gambar 5.2 Tampilan Tabel Dokumen bag.1.....	97
Gambar 5.3 Tampilan Tabel Dokumen bag.2.....	97
Gambar 5.4 Tampilan Pop Up Threshod Awal.....	98
Gambar 5.5 Tampilan Pop Up Ubah Threshod	98
Gambar 5.6 Tampilan Pop Up Proses <i>Training (Preprocessing)</i>	99
Gambar 5.7 Tampilan Pop Up Proses <i>Training</i> (Klasifikasi)	99
Gambar 5.8 Tampilan <i>Testing</i> Hasil Positif	100
Gambar 5.8 Tampilan <i>Testing</i> Hasil Negatif.....	100
Gambar 6.1 Diagram Nilai Akurasi pada tiap Pengujian.....	102

DAFTAR KODE PROGRAM

Kode Program 5.1 M_training_data.php	44
Kode Program 5.2 C_training_data.php.....	47
Kode Program 5.3 modelling_data.php – get data tabel.....	49
Kode Program 5.4 Text_processing.php.....	49
Kode Program 5.5 M_processing.php.....	52
Kode Program 5.6 processing.php.....	55
Kode Program 5.7 Text_processing.php - function term_frequency().....	58
Kode Program 5.8 processing.php - function term_frequency(data).....	59
Kode Program 5.9 Text_processing.php - get_entropy().....	60
Kode Program 5.10 processing.php – get_entropy(data).....	61
Kode Program 5.11 Text_processing.php - get_entropy_partisi_P_N().....	62
Kode Program 5.12 processing.php - get_entropy_partisi_P_N(entropy_total, data).....	66
Kode Program 5.13 Text_processing.php – info_gain().....	67
Kode Program 5.14 processing – info_gain(entropy_total, data, data_entropy).....	68
Kode Program 5.15 Text_processing.php – frekuensi_term2().....	69
Kode Program 5.16 processing.php – frekuensi_term2(infogain, data).....	70
Kode Program 5.17 Naive_bayes.php - function likelihood().....	72
Kode Program 5.18 processing.php – likelihood (info_gain, data, term_frequency2).....	75
Kode Program 5.19 Naive_bayes.php - simpan_likelihood().....	76
Kode Program 5.20 processing - simpan_likelihood(likelihood, id)	77
Kode Program 5.21 M_naivebayes.php – simpan_likelihood(\$data)	78
Kode Program 5.22 Naive_bayes.php - hapus_likelihood_get_prior().....	78
Kode Program 5.23 M_Naive_bayes - get_prior()	79
Kode Program 5.24 Tester.php.....	81
Kode Program 5.25 data_tester.php.....	84
Kode Program 5.26 Pengujian.php.....	87
Kode Program 5.27 M_pengujian.php.....	92
Kode Program 5.28 pengujian.php.....	93

DAFTAR LAMPIRAN

Data Latih Kelas Positif.....	114
Data Latih Kelas Negatif.....	118
Kamus Alay (Normalisasi Bahasa).....	122
Hasil nilai akurasi - nilai true positif, true negatif, false positif, false negatif K3.....	125
Hasil nilai akurasi - nilai true positif, true negatif, false positif, false negatif K5.....	126
Daftar stoplist / stop Removal.....	127
Surat Permohonan menjadi pakar untuk validasi data.....	128



BAB 1 PENDAHULUAN

1.1 Latar belakang

Perkembangan internet sudah sangat pesat dalam mempengaruhi media dan komunikasi, hal ini dibuktikan dengan penggunaan jejaring sosial yang semakin bertambah. Para konsumen memiliki aktifitas yang sangat padat dalam memposting berita, opini, maupun membalas komentar dan pesan. Salah satu jejaring sosial yang sangat populer adalah twitter, menurut data yang berupa infografis pada tahun 2014 – 2015, twitter memiliki 302 juta pengguna yang aktif dan 80% aksesnya berasal dari pemakaian perangkat mobile. Pengguna dengan usia sekitar 18-29 tahun mencapai 39% dari jumlah angka pengguna aktif yang sudah dipaparkan, sedangkan untuk usia 30 - 49 tahun sekitar 25% (Grafelly, 2015).

Twitter merupakan sebuah *microblog*, dikarenakan user dapat mengirimkan maksimal 140 karakter untuk di posting. Postingan tersebut dapat berisikan komentar maupun pujian pada suatu hal (Saifinnuha, 2013). Dari melihat cara kerja sistem dari jejaring ini sangat memungkinkan dapat mengirim lebih dari 200 juta *tweet* tiap harinya. Hasil review menurut MIT Technology, dalam pengiriman *tweet* terbanyak di seluruh dunia, Negara Indonesia menempati posisi ketiga dengan jumlah berkisar 1 Milyar *tweet* setelah Amerika yang memperoleh jumlah 3,7 Milyar dan Jepang dengan jumlah 1,8 Milyar pada tahun 2013. Bahkan untuk kota Jakarta mendapat gelar sebagai *Twitter Capital City*, dikarenakan memiliki jumlah terbanyak dan teraktif di dunia (Saifinnuha, 2013). Sedangkan berdasarkan media informasi CNN Indonesia, data yang diperoleh, sekitar 4,1 juta *tweet* yang berasal dari Indonesia, 77 persen pengguna twitter di Indonesia setiap harinya. Dari 77 persen, 54 persennya diantaranya melakukan 2 *tweet* setiap harinya (Maulana, 2016). Pada sumber terbaru, Kuartal I di tahun 2017 Pengguna Twitter bertambah hingga 6 % atau 9 juta, sehingga secara keseluruhan jumlah Pengguna Twitter mencapai sekitar 328 juta (tekno.kompas; 2017; *Pengguna Twitter Naik 9 Juta*; <https://tekno.kompas.com/read/2017/04/27/10230077/pengguna.twitter.naik.9.juta.penambahan.tertinggi.sejak.2015>; diakses tanggal 18 Maret 2018).

Adapun penjelasan yang lain (Capece, et al., 2009) twitter merupakan jejaring yang memiliki peran besar dalam mempengaruhi aspek dari segi sosial, komunikasi, psikologi, marketing, dan politik. Hal ini terjadi karena tidak sedikit postingan yang berisi mengenai opini masyarakat terhadap kejadian peristiwa, fenomena, atau bahkan pada suatu hal yang bersifat benda seperti produk suatu individu atau perusahaan. Para pengguna aktif twitter yang mengirimkan komentar atau opini secara tidak langsung akan menjadi sebuah review terhadap penilaian pada suatu produk tersebut, dan review tersebut dapat memberikan impact pada produk dimana akan meningkatkan nilai pasarnya atau malah sebaliknya.

Sektor produk yang diminati banyak orang saat ini adalah produk kecantikan dan perawatan kulit, salah satunya adalah produk dari Mustika Ratu. Hal ini didasari dari penjualan yang diperoleh mencapai milyaran rupiah. Namun dilihat dari segi pertumbuhan penjualan perusahaan yang didirikan oleh Hj. DR BRA Mooryati Soedibyo, SS., M.Hum. saat ini mengalami penurunan pendapatan pada kuartal ketiga 2017 sebesar 8,8% atau berkisar 236 miliar padahal pada kuartal tiga tahun lalu Mustika Ratu memperoleh penjualan bersih senilai 259 miliar (industri.kontan; 2017; *Mustika Ratu Mencetak Laba di Kuartal 3*; <https://industri.kontan.co.id/news/mustika-ratu-mencetak-laba-rp-15-miliar-di-q3>; diakses tanggal 18 Maret 2018). Oleh sebab itu dibutuhkan penanganan dan tindakan awal seperti pengukuran citra produk saat ini dengan proses analisis yang tepat untuk mengetahui penyebab dan dicarikan solusinya lebih lanjut, mengingat perusahaan Mustika Ratu adalah salah satu perusahaan yang didirikan oleh orang Indonesia.

Analisis sentimen atau disebut juga opinion mining adalah suatu proses dalam menyajikan informasi hasil dari olahan suatu sistem yang digunakan untuk mengklasifikasikan antara kategori positif dan negatif pada keseluruhan sentimen yang dinyatakan dalam tiap dokumen (Hidayat, 2015). Di dalam analisis sentimen ini akan dilakukan serangkaian proses untuk memahami, mengekstrak, dan mengolah data tekstual yang terkandung dalam kalimat opini, yang kemudian dikembangkan menjadi sebuah informasi sentimen yang dibutuhkan (Saifinnuha, 2013). Tujuan dari penggunaan analisis sentimen adalah untuk mengekstraksi atribut pada dokumen atau teks berisi komentar untuk mengetahui ekspresi didalamnya guna digolongkan menjadi positif dan negatif. Capaian kutub yang dimasukkan tidak harus selalu positif dan negatif tapi bisa dalam banyak hal selama memiliki kata sifat yang berlawanan, atau bisa ditambahkan kluster atau golongan baru sesuai kebutuhan (Liu, 2012). Melihat tujuan dari analisis sentiment akan sangat memungkinkan dalam mengatasi permasalahan produk kecantikan mustika ratu, untuk analisis kecenderungan pasar, kajian ulang terhadap kualitas produk dalam pandangan publik, analisis peningkatan citra dan pendapatan suatu Perusahaan khususnya Mustika Ratu. Sistem dapat diarahkan untuk proses menganalisis dan klasifikasi data atau komentar - komentar ataupun sentimen pengguna twitter terhadap pendapat atau pandangannya terkait postingan mengenai suatu produk tertentu yang sedang dipasarkan, guna mendapatkan hasil penilaian brand dan kecenderungan pasar yang bersifat positif maupun negatif.

Salah satu metode yang digunakan untuk proses pengklasifikasian dalam sentimen analisis adalah *Naïve Bayes Classifier*. Pemilihan Metode *Naïve Bayes Classifier* didasari beberapa alasan seperti kemudahan dalam implementasinya, dan memiliki performa cepat dalam melatih, serta dapat digunakan pada dataset yang lebih besar. Salah satu penelitian sebelumnya (pamungkas, et all 2015) yang berjudul Analisis Sentimen pada Sosial Media Twitter Menggunakan *Naive Bayes Classifier* terhadap Kata Kunci "Kurikulum 2013" melakukan klasifikasi pada data sentimen Twitter dengan kata kunci "Kurikulum 2013". Pengklasifikasian tersebut menggunakan machine learning yang dilengkapi algoritma dengan

memanfaatkan nilai probabilitas dari setiap data tweet yang kemudian nilainya dibandingkan dalam sebuah table Vmap positif, Vmap netral, dan Vmap negatif. Hasil klasifikasi pada data tweet tertentu akan dikategorikan pada kelas positif ketika ketika Vmap positif lebih besar dari Vmap negatif, dan begitu pun sebaliknya. Hasil dari penelitian tersebut memberikan nilai akurasi sebesar 91% untuk 1000 data latih yang diberikan, dan akan semakin akurat lagi jika data latih yang digunakan semakin banyak dalam data pengetahuan, namun untuk analisis selain Bahasa Indonesia (bahasa asing dan bahasa daerah) masih berjalan kurang maksimal. Disebutkan pada beberapa penelitian yang sudah dilakukan dalam perbandingan 3 metode, yakni Naïve Bayes, Decision Tree, dan Neural Network guna mengatasi teknik pembelajaran mesin untuk permasalahan analisis sentimen, 3 metode tersebut dilatih dengan menggunakan 105 data dan di uji 521 data, secara keseluruhan terdiri dari 3900 fitur / kata yang unik. Hasil yang didapatkan Metode Naive Bayes Classifier mencapai akurasi (97,89%), presisi (99,20%), recall (98,6%) dan F-Measure (98,90%). Metode Decision Tree mencapai hasil akurasi (94,85%), presisi (98,31%), recall (95,90%), dan F-Measure (97,09%.) Sedangkan pada Neural Network, diproses dengan 3900 input, 3900 midnodes dan 1 output, dengan hasil pada peringkat terakhir di semua metrik dan waktu eksekusinya. Meskipun Metode Decision Tree mengungguli Naive Bayes Classifier dalam kecepatan eksekusi dan Recall value (sebesar 0,67%). Namun, Naive Bayes Classifier mencapai Akurasi dan Presisi yang lebih tinggi, dan yang paling penting, nilai F-Measure yang secara keseluruhan merupakan hasil yang sangat menjanjikan (Xhemali, et al. 2009).

Namun pada sisi bagian yang lain dalam proses klasifikasi teks terdapat permasalahan yang sangat umum dikarenakan dimensi tinggi dari ruang fitur. Tingkat ketinggian yang terjadi pada ruang fitur ini disebabkan dari domain teks yang setiap hari bertambah terus hingga memiliki ribuan fitur, sehingga menyebabkan pemrosesan yang lambat, selain itu beberapa fitur tidak relevan dan bermanfaat atau pun dapat mengurangi tingkatan akurasi fitur yang sesuai pada klasifikasi teks yang diinginkan (Chen et al 2009). Kemudian dalam perkembangan klasifikasi teks agar dapat ditingkatkan dalam memperoleh nilai akurasi yang baik dalam model yang lebih rumit pada analisis sentimen, maka dapat diterapkan proses seleksi fitur dan menghilangkan noise yang tidak diharapkan. Pemilihan fitur yang dimaksud adalah sebuah metode untuk meminimalisir kelemahan dari metode *Naïve Bayes Classifier*, dengan mengurangi jumlah data yang dianalisa dan mengidentifikasi fitur-fitur yang masih sesuai untuk bahan pertimbangan dalam proses pembelajaran mesin (Moraes, aliati, & Neto, 2013). Penelitian yang sudah dilakukan terkait pernyataan sebelumnya (Muthia, 2016) berjudul Opinion Mining pada Review Buku Menggunakan Algoritma Naïve Bayes. Eksperimen dilakukan dengan menggunakan metode *Naïve Bayes* saja, setelah itu dilakukan dengan menggunakan metode Naïve Bayes yang digabungkan dengan metode pemilihan fitur *Information Gain* dan Algoritma *Genetic*. Keduanya diukur dengan menggunakan *Matrix Confusion* untuk mendapatkan hasil akurasi dari setiap eksperimen, kemudian dibandingkan keduanya. Dinyatakan bahwa klasifikasi

pada data review buku menjadi kategori positif dan kategori negatif. Akurasi Naïve Bayes sebelum menggunakan penggabungan metode pemilihan fitur mencapai 78,50%. Sedangkan menggunakan penggabungan metode pemilihan fitur, terdapat perubahan yang meningkat menjadi 84,50%. Sehingga dapat dikatakan penggabungan dari metode pemilihan fitur dapat meningkatkan akurasi dengan selisih 6% lebih tinggi.

Metode naïve bayes dan *Information Gain* sebagai pemilihan fiturnya diharapkan akan mengklasifikasikan komentar para pengguna twitter tersebut dengan akurasi yang tergolong baik, kemudian dihitung dan diolah untuk mempresentasikan tingkatan nilainya sehingga dapat mempengaruhi eksistensi produk dan nilai jual / nilai pasarnya. Berdasarkan latar belakang diatas, maka penulis terdorong untuk menuliskan skripsi yang berjudul **“Penerapan Analisis Sentimen untuk Menilai Suatu Produk pada Twitter Berbahasa Indonesia dengan Metode Naïve Bayes Classifier dan Informaton Gain”**.

1.2 Rumusan masalah

Berdasarkan latar belakang dari penelitian tugas akhir yang berjudul Penerapan Analisis Sentimen untuk Menilai Suatu Produk pada Twitter Berbahasa Indonesia dengan Metode *Naïve Bayes Classifier* dan *Informaton Gain*, maka ada beberapa masalah yang diidentifikasi sebagai berikut :

1. Bagaimana menerapkan Metode *Naïve Bayes Classifier* yang dikombinasikan dengan metode *Information Gain* untuk analisis sentimen pada twitter guna menilai suatu produk.
2. Bagaimana tingkat akurasi pada poses klasifikasi sentimen positif dan negatif dengan menggunakan metode *Naïve Bayes Classifier* yang ditambahkan dengan metode *Information Gain*.

1.3 Tujuan

Berdasarkan latar belakang penelitian dan rumusan masalah yang sudah dipaparkan, maka tujuan dari penelitian ini adalah :

1. Menerapkan analisis sentimen pada data sentimen twitter berbahasa indonesia agar mendapatkan penilaian dari suatu produk yang mengarah pada kelas positif atau kelas negatif.
2. Seberapa baik tingkat akurasi pada klasifikasi sentimen dengan menggunakan metode *Naïve Bayes Classifier*. yang ditambahkan dengan metode *Information Gain*.

1.4 Manfaat

Manfaat yang diinginkan dari penelitian ini adalah memudahkan para stakeholder seperti perusahaan untuk mengetahui nilai jual dan kecenderungan pasar yang terjadi pada produk sebagai evaluasi kualitas produk dan pengembangan strategi marketing. Serta sebagai rekomendasi untuk menaruh minat dan pengetahuan terhadap produk bagi masyarakat yang belum paham.

1.5 Batasan masalah

Batasan masalah dari penelitian ini sebagai berikut :

1. Penelitian ini hanya berfokus pada data sentimen twitter yang mengandung review terhadap produk tertentu alat kosmetik & kecantikan, khususnya pada produk Mustika Ratu.
2. Sistem yang dimaksudkan adalah sistem untuk menilai produk tertentu berbasis web dengan menggunakan PHP dan struktur bangun MVC (Model, View, Controller).
3. *Tweet* yang akan diproses adalah *tweet* yang berbahasa Indonesia.
4. Penentuan Data sentiment pada twitter yang akan dipakai dilakukan oleh penulis sendiri sesuai kebutuhan, secara khusus pada pemilihan isi dokumen, penggunaan dan pengukuran jumlah jenis kata pada dokumen berkelas positif dan dokumen berkelas negatif.
5. Jumlah Dokumen Positif dan Jumlah Dokumen Negatif seimbang, dengan kata lain perbandingan jumlahnya sama untuk pertimbangan mengatasi kecenderungan nilai probabilitas tidak seimbang antara probabilitas positif maupun probabilitas negatif.

1.6 Sistematika pembahasan

Sistematika Pembahasan yang ada pada penelitian ini adalah sebagai berikut :

a. BAB 1 PENDAHULUAN

Bagian ini berisikan latar belakang dimana akar dari penulis melakukan penelitian ini, kemudian berisi rumusan masalah untuk identifikasi masalah yang akan dipecahkan, tujuan, manfaat, batasan masalah, dan sistematika pembahasan dari penelitian ini.

b. BAB 2 LANDASAN KEPUSTAKAAN

Landasan Kepustakaan berisi tentang teori, konsep, dan kajian-kajian penelitian sebelumnya yang relevan dengan penelitian ini. Penulisan kajian tersebut ditata sedemikian rupa sehingga membentuk Skema teoritis untuk menyelesaikan permasalahan dari masalah penelitian ini.

c. BAB 3 METODOLOGI

Berisikan penjelasan terkait metodologi yang digunakan untuk menyelesaikan masalah dalam menilai suatu produk dengan memanfaatkan analisis sentimen menggunakan metode *Naïve Bayes Classifier* dan *Information Gain*.

d. BAB 4 ANALISIS DAN PERANCANGAN

Menyajikan analisis dan perancangan sistem metode klasifikasi *Naive Bayes* dan *Information Gain*, perancangan pengujian dan manualisasi untuk menyajikan model hasil dari metode yang digunakan,

e. BAB 5 IMPLEMENTASI

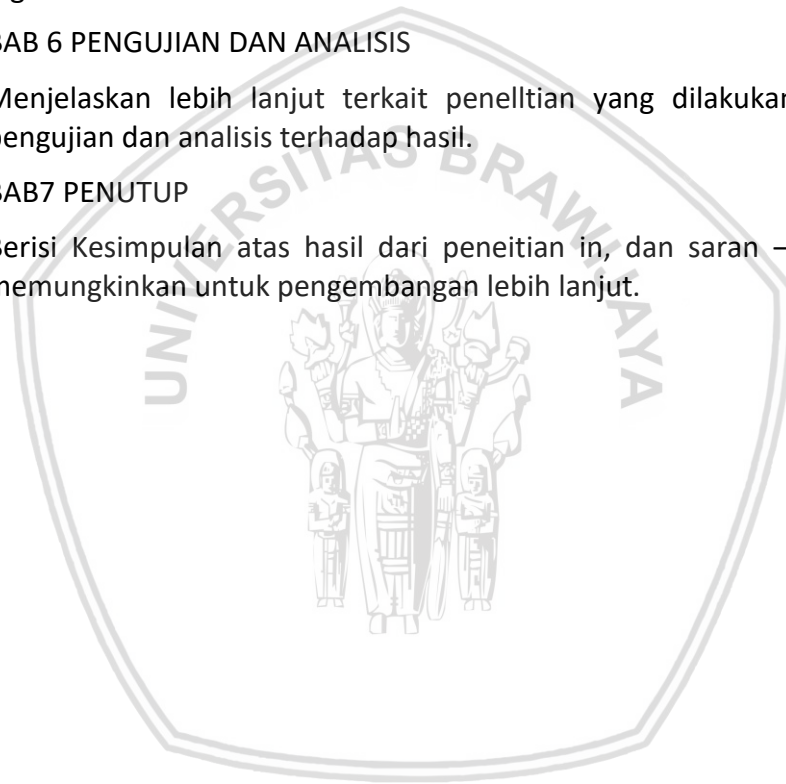
Pada bagian ini, menjelaskan proses implementasi dan algoritma yang digunakan.

f. BAB 6 PENGUJIAN DAN ANALISIS

Menjelaskan lebih lanjut terkait penelitian yang dilakukan, hasil dari pengujian dan analisis terhadap hasil.

g. BAB 7 PENUTUP

Berisi Kesimpulan atas hasil dari penelitian ini, dan saran – saran yang memungkinkan untuk pengembangan lebih lanjut.



BAB 2 LANDASAN KEPUSTAKAAN

Bab ini berisi Kajian pustaka dan penjelasan tentang teori dasar yang diperlukan untuk membangun sistem yang menerapkan analisis sentimen pada twitter berbahasa Indonesia untuk menilai produk dari suatu perusahaan. Kajian Pustaka membahas penelitian – penelitian sebelumnya yang berhubungan dengan masalah penelitian yang diusulkan dan untuk memudahkan dalam proses penyusunan.

Sedangkan terori dasar dijelaskan untuk memberikan pemahaman dalam menyelesaikan permasalahan pada penelitian ini. Adapun teori dasar yang dibahas, antara lain Text Mining, Analisis sentimen, metode yang digunakan untuk implementasiannya (*Naïve Bayes Classifier* dan *Information Gain*), dan bagian dari twitter dalam pengambilan data yang diperlukan, serta Penilaian Produk.

2.1 Kajian Pustaka

Isi Kajian Pustaka ini akan membahas penelitian yang berjudul **“Penerapan Analisis Sentimen untuk Menilai Suatu Produk pada Twitter Berbahasa Indonesia dengan Metode *Naïve Bayes Classifier* dan *Information Gain*”** dan penelitian-penelitian sebelumnya yang relevan dengan penelitian yang dilakukan oleh penulis. Penelitian yang dilakukan (Retnawiyati et al., 2015) terkait dengan analisis sentimen terhadap produk dengan mengelolah data twitter menggunakan *Naïve Bayes Classifier* dimana diklasifikasikan menjadi opini positif, negatif, dan netral, kemudian disimpulkan metode tersebut dapat memberikan data yang cukup akurat selama data latih yang digunakan lebih banyak. Sedangkan penelitian analisis sentimen selanjutnya (Muthia, 2016) menerapkan metode *Naïve Bayes Classifier* dengan ditambahkan metode *Genetic Algoritma* dan pemilihan fitur yakni *Information Gain* guna meningkatkan nilai akurasi pengklasifikasian terhadap data review buku .

Penelitian yang pertama berjudul Analisis Sentiment pada Sosial Media Twitter Menggunakan *Naïve Bayes Classifier* terhadap Kata Kunci “Kurikulum 2013” (pamungkas, et all 2015) melakukan klasifikasi pada Data sentiment Twitter dengan kata kunci “Kurikulum 2013”. Pengklasifikasian tersebut menggunakan machine learning yang dilengkapi algoritma dengan memanfaatkan nilai probabilitas dari setiap data twet yang kemudian nilainya dibandingkan dalam sebuah table Vmap positif, Vmap netral, dan Vmap negatif. Hasil dari penelitian tersebut memberikan nilai akurasi sebesar 91% untuk 1000 data latih yang diberikan, dan akan semakin akurat lagi jika data latih yang digunakan semakin banyak dalam data pengetahuan, namun untuk analisis selain Bahasa Indonesia (bahasa asing dan bahasa daerah) masih berjalan kurang maksimal.

Penelitian yang kedua berjudul Real Time Sentiment Analysis of Tweet Using *Naïve Bayes* (Goel Ankur, et al. 2016) mengklasifikasikan data training 140

sentimen dari database twitter menggunakan *Naïve Bayes Classifier* menjadi kelas positif dan negatif. Ketika dilatih dengan 1,6 juta tweet dari 140 dataset sentiment dan diuji dengan 100 tweet terbaru, hasil penelitian menunjukkan akurasi 58,40%. Sedangkan dengan menggunakan sentiwordnet dan *Naïve Bayes*, maka nilai akurasi dapat ditingkatkan hingga batas tertentu seperti yang diusulkan.

Penelitian yang ketiga, berjudul *Opinion Mining pada Review Buku Menggunakan Algoritma Naïve Bayes* (Muthia, 2016). Eksperimen dilakukan dengan menggunakan metode Naïve Bayes saja, setelah itu dilakukan dengan menggunakan metode Naïve Bayes yang digabungkan dengan metode pemilihan fitur *Information Gain* dan *Algoritma Genetic*. Keduanya diukur dengan menggunakan *Matrix Confusion* untuk mendapatkan hasil akurasi dari setiap eksperimen, kemudian dibandingkan keduanya. Dinyatakan bahwa klasifikasi pada data review buku menjadi kategori positif dan kategori negatif. Akurasi Naïve Bayes sebelum menggunakan penggabungan metode pemilihan fitur mencapai 78,50%. Sedangkan menggunakan penggabungan metode pemilihan fitur, terdapat perubahan yang meningkat menjadi 84,50%

Penelitian yang keempat dilakukan oleh Joang Ipmawati dkk dengan judul *Komparasi Teknik Klasifikasi Teks Mining pada Analisis Sentimen*, membandingkan beberapa algoritma klasifikasi dalam melakukan teks mining diantaranya *Support Vector Machine (SVM)*, *Naïve Bayesian Classification (NBC)* dan *K-Nearest Neighbor (K-NN)*. Ketiga algoritma ini akan dikomparasi untuk mengetahui performa yang baik dalam hal akurasi untuk dua dataset yang berbeda yaitu imdb review film dan sentimen twitter. Hasil dari komparasi menunjukkan SVM memperoleh hasil yang baik dalam akurasi pada data imdb review film 78,55% dan pada dataset twitter 72%. Sama halnya dengan NBC yang memperoleh akurasi pada data twitter 78,55% tetapi berbeda pada data imdb review film 67,33%. Hasil F-measure review film menunjukkan SVM dan NBC memperoleh hasil yang sama yaitu 0,785 dan untuk hasil AUC, NBC mengungguli hasil 0,869, SVM memperoleh hasil 0,786 sedangkan KNN memperoleh hasil 0,572. F-measure untuk twitter SVM lebih unggul memperoleh hasil 0,720 dan NBC memperoleh hasil 0,673 sedangkan K-NN 0,545. Dan untuk hasil AUC, sama seperti dataset imdb, pada dataset twitter ini NBC juga mengungguli SVM dan K-NN. AUC untuk NBC memperoleh hasil 0,735, SVM memperoleh hasil 0,658 dan K-NN memperoleh hasil 0,618.

Tabel 2.1 Tabel Kajian Pustaka

Tahun	Judul	Peneliti	Objek	Hasil Penelitian
2015	Analisis Sentiment pada Sosial Media Twitter Menggunakan	Dyarsa Singgih Pamungkas, Noor Ageng Setyanto, dan	Data Sentimet Twitter yang mengandung kata kunci Kurikulum	Hasil penelitian berupa sebuah sistem yang dapat

	<i>Naive Bayes Classifier</i> terhadap Kata Kunci "Kurikulum 2013"	Erlin Dolphina	2013	mengklasifikasi sentimen secara otomatis dengan hasil pengujian 3000 data latih dan 100 tweet data ujicoba mencapai 91 %
2016	<i>Real Time Sentiment Analysis of Tweet Using Naïve Bayes</i>	Ankur Goel, Jyoti Gautam dan Sltesh Kumar	Data Sentimen pada Twitter	Bahwa proses klasifikasi dengan naïve bayes pada 100 tweet terbaru, akurasi menunjukkan 58,40%.
2016	Opinion Mining pada Review Buku Menggunakan Algoritma Naïve Bayes	Dinda Ayu Muthia	Data Review Buku	Akurasi Naïve Bayes sebelum menggunakan penggabungan metode pemilihan fitur mencapai 78,50%. Sedangkan menggunakan penggabungan metode pemilihan fitur 84,50%.
2017	Komparasi Teknik Klasifikasi Teks Mining pada Analisis Sentimen	Joang Ipmawati, Kusrini, dan Emha Taufiq Luthfi	Dataset Review film IMDB dan data sentiment twitter	hasil accuracy Support Vector Machines dengan Naïve Bayes Classification = 78,55% sedangkan pada dataset

				twitter SVM memperoleh hasil 72%, sedangkan pada NBC dan KNN sebesar 67,33% dan 56,83%
--	--	--	--	--

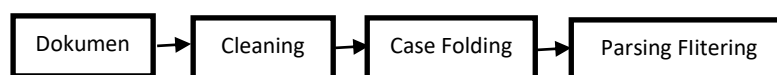
2.2 Text Mining

Teks Mining adalah kegiatan penambangan teks yang dilakukan oleh komputer untuk mendapatkan sesuatu yang baru, yang tidak diketahui atau menemukan kembali informasi yang tersirat secara implisit, yang bersumber pada informasi hasil ekstraksi dari data teks yang berbeda-beda (Feldman & Sanger, 2006).

Secara umum, proses yang dilakukan pada penambangan teks adalah dengan otomatisasi perangkuman, kategorisasi dokumen, penggugusan teks, dll (Turban, dkk, 2005). Jenis masukan atau inputan dari penambangan teks ini berupa data yang tidak terstruktur, sehingga ada 2 tahap yang dilakukan dalam prosesnya, yakni penerapan terstruktur pada sumber data teks dan ekstraksi informasi, ekstraksi juga dilakukan pada pengetahuan yang relevan dengan data terstruktur ini dengan menggunakan teknik dan alat yang sama dalam penambangan data (Ipmawati et, al. 2017). Sedangkan penjelasan yang serupa (Berry & Kogan, 2010) Teks Mining (Penambangan Teks) secara umum adalah *text preprocessing* dan *feature selection*.

2.2.1 Text Preprocessing

Text Preprocessing adalah bagian awal dari proses teks mining, dimana pada tahap ini dilakukan rutinitas dalam penyediaan data yang sudah siap digunakan pada operasi *knowledge discovery* sistem teks mining (Feldman & sanger, 2006). Adapun 2 proses yang berjalan pada tahap Text Preprocessing, yang pertama yang dikenal dengan istilah Tokenizing yaitu proses penguraian pada kalimat menjadi dalam bentuk kata dengan menghilangkan delimiter yang berupa tanda titik (.), koma (,), karakter putih / spasi, dan karakter angka yang ada pada kata tersebut. Sedangkan yang kedua dikenal dengan *toLowerCase*, yaitu mengubah karakter huruf yang besar / kapital menjadi huruf kecil (Weiss, et al., 2005). Berikut Gambar 2.1 jalur dari proses text preprocessing :



Gambar 2.1 Tahapan Proses Text Preprocessing

2.2.2 Feature Selection

Tahap seleksi fitur (*feature selection*) bertujuan untuk mengurangi ruang lingkup dimensi dari beberapa kumpulan teks, atau menghapus kata-kata yang dianggap tidak penting agar pemrosesan dokumen dalam klasifikasi lebih akurat dan efektif (Feldman & Sanger, 2006). Tahapan ini akan menghilangkan stopwords dan stemming pada kata yang berimbuhan (Berry & Kogan, 2010)

Stopword berupa kosakata yang merupakan ciri unik dari dokumen (Dragut, et al., 2009). Disamping stopwords sudah dijalankan, proses selanjutnya akan dilakukan stemming. Berikut Tabel 2.1 merupakan stopwordslist dengan bahasa Inggris dari referensi di internet.

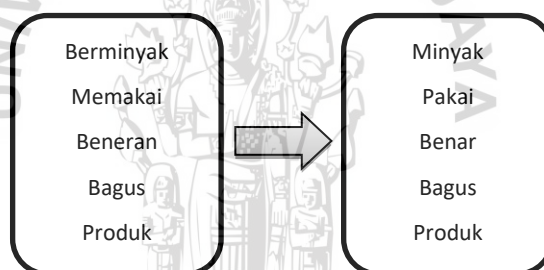
Tabel 2.2 Tabel StopWord List

	STOP WORD LIST
A	a, about, again, all, almost, also, although, always, among, an, and, another, any, are, as, at
B	be, because, been, before, being, between, both, but, by
C	Can, could
D	did, do, does, done, due, during
E	Each, either, enough, especially, etc
F	for, found, from, further
H	had, has, have, having, here, how, however
I	i, if, in, into, is, it, its, itself
J	Just
K	Kg, km
M	made, mainly, make, may, mg, might, ml, mm, most, mostly, must
N	nearly, neither, no, nor
O	obtained, of, often, on, our, overall
P	perhaps, permitted
Q	Quite
R	rather, really, regarding
S	seem, seen, several, should, show, showed, shown, shows, significantly, since, so, some, such
T	than, that, the, their, theirs, them, then, there, therefore, these, they, this, those, through, thus, to
U	upon, use, used, using

V	various, very
W	was, we, were, what, when, which, while, with, within, without, would

Sumber: NCBI (2017)

Tiap kata yang berbahasa inggris diatas bisa diterjemahkan menjadi bentuk kata bahasa indonesianya, supaya dapat digunakan dalam menghapus kata yang tidak perlu pada dokumen bahasa indonesia. Saat proses filtering dengan stoplist selesai, tindakan selanjutnya adalah proses stemming, yaitu sebuah proses untuk menguraikan kata dalam bentuk yang bervariasi menjadi bentuk awalnya atau kata dasarnya (Tala, 2003). Proses ini bertujuan untuk menghilangkan imbuhan seperti prefix, sufiks, dan konfiks yang ada pada setiap kata, sehingga kata yang memiliki kata dasar yang sama meskipun dengan bentuk variasi yang berbeda sistem akan menyimpan kata dasarnya cukup sekali saja. Hal ini akan mempengaruhi efektifitas kinerja dari sistem. Disebutkan dalam <https://github.com/jsastrawi/jsastrawi>, Sastrawi Stemmer merupakan salah satu library untuk proses stemming yang kerap digunakan karena sederhana dan mudah untuk digunakan. Pada penelitian ini digunakan Sastrawi Stemmer yang berupa sebuah *library* yang akan diinputkan kedalam program untuk mendapatkan kata dasar pada setiap dokumen tweet. Gambar 2.2 merupakan langkah stemming.



Gambar 2.2 Stemming

2.3 Analisis Sentimen

Analisis Sentimen adalah salah satu cabang penelitian text mining. Analisis sentimen berkaitan dengan bidang yang lebih luas seperti pengolahan bahasa alami, komputasi linguistik, dan text mining untuk tujuan menganalisis, sentimen, pendapat, sikap, evaluasi, penilaian dan emosi seseorang berkenaan pada topik, produk, layanan, organisasi, atau kegiatan tertentu (Liu, 2012). Sedangkan arti dari sentimen menurut KBBI (Kamus Besar Bahasa Indonesia) adalah suatu pendapat atau pandangan yang diutarakan dengan perasaan yang berlebihan, atau dengan kata lain bertentangan dengan pertimbangan pikiran, hasil dari sentimen dapat berupa sentimen yang bersifat positif maupaun sentimen yang bersifat negatif.

2.3.1 Opinion Lexicon

Opinion Lexicon diperlukan dalam analisis sentimen ketika dilakukan pengklasifikasian dokumen yang berupa opini yang mengandung sentimen dari

dokumen yang sedang diproses. Proses yang terjadi melibatkan Opinion Lexicon dalam menganalisis dokumen sehingga dapat disimpulkan dokumen yang mengandung sentimen dan sebaliknya. Opinion Lexicon berupa kata-kata yang mengandung sentimen. Sedangkan kata itu sendiri terdiri sebagai *sentiment words*, *opinion-bearing*, dan *polar words*.

Beberapa contoh kata yang mengandung opini bersentimen positif, seperti “cantik”, “bagus”, “luar biasa”, dan “indah”. Sedangkan kata yang mengandung opini bersentimen negatif, “lemah”, “payah”, dan “jelek” (Liu, 2012).

Adapun 2 macam penggunaan kata yang mengandung opini, antara lain opini langsung dan perbandingan. Opini langsung adalah opini yang mengekspresikan sentiment terhadap target berupa sebuah obyek, seperti topik, orang, atau produk didalam sebuah dokumen. Contoh : “Kualitas gambar dari kamera ini luar biasa”. Sedangkan opini perbandingan adalah opini yang membandingkan dari sifat kemiripan atau perbandingan dari obyek yang lebih dari satu Contoh L “Kamera merek X lebih murah daripada merek Y”. (Pang & Lee, 2008).

2.4 Information Gain

Information Gain adalah suatu nilai yang digunakan untuk menyatakan kesesuaian kata atau fitur terhadap kriteria yang ditetapkan pada *machine learning* (Mittchell, 1997). *Information gain* merupakan salah satu tahapan seleksi fitur juga yang digunakan untuk mengurangi keberadaan kata / fitur yang tidak relevan. Penelitian yang dilakukan oleh Rasmawati dan Bhaskaran melakukan studi perbandingan lima teknik seleksi fitur yang salah satunya adalah *Information gain*, kemudian digabungkan masing-masing dengan 4 algoritma klasifikasi pada data mining. Hasil eksperimen yang diperoleh adalah teknik seleksi fitur *Information Gain* menunjukan hasil terbaik (Ramaswami & Bhaskaran, 2009). Untuk mendapatkan nilai *Information Gain*, harus mencari nilai Entropy. Entropi digunakan untuk menentukan tingkat *informative* suatu inputan atribut untuk menghasilkan sebuah atribut (Kamagi, 2014). Entropi juga bisa disebut sebagai parameter untuk mengukur tingkat keberagaman (heterogenitas) atas beberapa kumpulan data.

Berikut Persamaan (2.1) untuk mencari nilai Entropy :

$$Info(D) = - \sum_{i=1}^c P(i) \log_2 P(i) \quad (2.1)$$

Keterangan pada Rumus 2.1 :

c : jumlah nilai pada atribut target (Jumlah kelas klasifikasi)

P(i) : proporsi sample pada kelas i dengan sample total

Setelah mendapatkan nilai entropi dari setiap fitur, maka dicari nilai entropy setelah pembobotan pada setiap fiturnya, Persamaan (2.2).

$$Info_A(D) = - \sum_{j=1}^v \frac{|D_j|}{|D|} Info D_j \quad (2.2)$$

Keterangan pada Persamaan (2.2) :

A : atribut

|D| : jumlah seluruh sampel data

|D_j| : jumlah sampel untuk nilai partisi j

v : suatu nilai yang mungkin untuk atribut A

Info (D_j) : Entropy untuk setiap partisi j

Nilai *Information Gain* didapatkan dari pengurangan nilai dari Persamaan (2.1) dengan nilai dari Persamaan (2.2), seperti di bawah berikut :

$$InfoGain(A) = - Info(D) - | Info_A(D) | \quad (2.3)$$

Penggunaan metode *Information Gain*, akan ditambahkan pada proses seleksi fitur untuk membantu dalam memaksimalkan pemilihan fitur yang ada dalam dimensi teks yang cukup luas / data yang digunakan.

2.5 TF Biner

Pada *Term Frequency* (tf), terdapat beberapa jenis formula yang dapat digunakan yaitu (Mandala, 2004):

1. tf biner (*binery tf*), term frekuensi ini hanya memperhatikan apakah suatu kata ada atau tidak dalam dokumen, jika ada diberi nilai 1, jika tidak diberi nilai 0.
2. tf murni (*raw tf*), nilai tf diberikan berdasarkan seberapa banyak jumlah kata yang muncul pada dokumen tersebut. Contoh : Jika muncul 7 kali maka kata tersebut akan bernilai 7.
3. tf logaritmik, hal ini untuk menghindari dominansi dokumen yang mengandung sedikit kata dalam *query*, namun mempunyai frekuensi yang tinggi.

$$tf = 1 + \log (tf) \quad (2.4)$$

4. tf normalisasi, menggunakan perbandingan antara frekuensi sebuah kata dengan jumlah keseluruhan kata pada dokumen.

$$tf = 0.5 + 0.5 \times tf \max tf \quad (2.5)$$

Dalam proses penelitian ini, tf yang digunakan adalah tf biner dimana hasil yang didapatkan berupa sebuah tabel vector yang dapat dikenal dengan sebutan *vector term presence* untuk digunakan dalam proses selanjutnya, yakni proses seleksi fitur.

2.6 Naïve Bayes Classifier

Naive Bayes Classifier merupakan metode pengklasifikasi sederhana berdasarkan penerapan teorema Bayes (Statistik Bayesian), biasanya digunakan dalam permasalahan klasifikasi teks. Misal sebuah ilustrasi data training yang dikategorikan menjadi beberapa k kategori $C_j = \{C_1, C_2, C_3, \dots, C_k\}$, dan *prior probability* untuk setiap kategori adalah $p(C_j)$, dimana $j = 1, 2, 3, \dots, k$.

Sedangkan untuk d_i disimbolkan sebagai koleksi data dimana $d_i = (w_1, \dots, w_j, \dots, w_m)$, dan fitur atau kata yang ada dalam dokumen berupa w_j , dimana $j = 1, 2, 3, \dots, m$, akan masuk ke dalam kategori C_j . Untuk mengklasifikasikan dokumen d_i , dilakukan dengan menghitung nilai probabilitas dari semua dokumen (*posterior probability*). Sehingga *posterior probability* suatu dokumen pada suatu kategori dapat dihitung melalui Persamaan (2.6),

$$p(C_j | d_i) = \frac{p(C_j | d_i) p(C_j)}{p(d_i)} \quad (2.6)$$

Kemudian untuk proses klasifikasi teks dilakukan memaksimalkan nilai dari Persamaan (2.6), dengan cara menghilangkan nilai penyebut $p(d_i)$ yang bernilai sama atau konstan, hal ini disebabkan semua kategori yang diberikan sama. Oleh sebab itu Persamaan (2.6) untuk menghitung nilai maksimal menjadi,

$$\sum_C^{C_j \max} p(C_j | d_i) = \sum_C^{C_j \max} p(d_i | C_j) p(C_j) \quad (2.7)$$

Jika dilihat dari hipotesa *Bayesian* sebelumnya, yang menyatakan bahwasannya setiap fitur atau kata $w_1, \dots, w_j, \dots, w_m$ dari $d_i = w_1, \dots, w_j, \dots, w_m$ adalah tidak saling terikat, sehingga distribusi probabilitas total adalah hasil perkalian (*product*) dari distribusi probabilitas tiap fitur atau kata, seperti Persamaan (2.8) berikut,

$$\begin{aligned} p(C_j | d_i) &= P(w_1, \dots, w_j, \dots, w_m | C_j) \\ &= \prod_{i=1}^m P(w_i | C_j) \end{aligned} \quad (2.8)$$

Dengan menggabungkan Persamaan (2.8) ke dalam Persamaan (2.7), maka menjadi Persamaan (2.9) seperti ini,

$$\sum_C^{C_j \max} p(C_j | d_i) = \sum_C^{C_j \max} p(C_j) \prod_{i=1}^m P(w_i | C_j) \quad (2.9)$$

Persamaan (2.9) inilah yang menjadi formula dalam Pengklasifikasian.

2.7 Twitter

Twitter merupakan situs web jejaring sosial berupa mikroblog yang dikelola oleh *Twitter Inc.*, sehingga dalam penggunaannya memungkinkan untuk mengirim dan membaca pesan, atau dikenal dengan istilah Tweets (*Twitter*, 2015). Para pengguna dapat mengirimkan 140 karakter untuk di post di twiiter, dari 140 karakter tersebut dapat berupa pujian atau komentar pada suatu hal. Bagian yang menjadi ciri khas dari jejaring ini adalah postingan berupa teks sebanyak 140 karakter dengan sebutan *Tweet* (baca : *twit*), yang bersifat public dan dapat dibatasi untuk dikirimkan kepada beberapa teman yang dikehendaki. Sedangkan sebutan *follower* diberikan kepada pengguna yang sudah menjadi teman sehingga dapat melihat *twit* pengguna lain dari halaman utaman mereka. Menurut (Capece, et al, 2009) twitter merupakan jejaring yang memiliki peran besar dalam mempengaruhi aspek dari segi sosial, komunikasi, psikologi, marketing, dan politik.

Seperti penjelasan sebelumnya twitter juga termasuk sebuah microblog yang merupakan alat komunikasi online yang memungkinkan para pengguna dapat memperbarui status tentang mereka yang sedang memikirkan atau melakukan sesuatu, seperti pendapat dan komentar pada suatu objek dan fenomena tertentu (Saifinnuha, 2013).

2.8 Penilaian Produk

Menurut Kotler (2005) menjelaskan bahwa produk merupakan segala sesuatu yang ditawarkan dalam memuaskan baik kebutuhan maupun keinginan. Produk diperlukan untuk mempertemukan hasil perusahaan dengan permintaan yang ada agar produk yang diperlukan oleh konsumen, memberikan kepuasan pada konsumen dan sekaligus menguntungkan perusahaan. Sedangkan untuk menilai suatu produk adalah kegiatan untuk mengamati dan menghasilkan sebuah keputusan atas dasar pengamatan tersebut yang dapat berupa positif maupun negatif. Salah satu cara adalah dengan sebuah review, dimana berisi tinjauan terhadap sebuah karya mengenai kelebihan dan kekurangannya. Karya tersebut dapat berupa film, buku, lagu, atau benda lain.

Sebuah produk memerlukan sebuah penilaian untuk sebagai tolak ukur atas kualitas produk itu sendiri dan sebagai sebab atas perilaku konsumen. Menurut Kotler (2007) perilaku konsumen adalah kegiatan-kegiatan individu yang secara langsung terlibat dalam mendapatkan dan mempergunakan barang-barang dan jasa-jasa, termasuk di dalamnya proses pengambilan keputusan pada persiapan dan penentuan kegiatan-kegiatan tersebut. (Dharmmerta dan Handoko, 2000). Pada dasarnya perilaku konsumen tersebut mengandung 2 (dua) elemen penting, yaitu : Proses pengambilan keputusan dan kegiatan fisik, semuanya melibatkan individu dalam menilai, mendapatkan dan mempergunakan barang-barang dan jasa-jasa ekonomis.

Di dalam pengambilan keputusan atas perilaku kosumen tersebut akan memepengaruhi daya jual produk dan nilai kualitasnya. Ketika konsumen

memutuskan sebuah produk memiliki nilai positif maka akan meningkatkan daya jual dan kualitasnya. Menurut Kotler (2008) semakin tinggi tingkat kualitas, maka akan semakin tinggi tingkat kepuasan yang dirasakan oleh konsumen, sehingga dari tingginya kualitas dari produk yang dirasakan konsumen, maka konsumen akan merekomendasikan produk terhadap orang lain. Hal ini pun akan mempengaruhi citra dari produk tersebut yang merupakan wajah dari produk itu sendiri dan sebagai kunci untuk memasarkan suatu produk.

2.9 Framework

Framework adalah kumpulan perintah atau fungsi yang sudah dikemas sedemikian rupa membentuk sebuah aturan-aturan yang saling berinteraksi dengan yang lain sehingga mempermudah dalam pembuatan website dengan cara mengikuti aturan yang ada pada framework tersebut, seperti penunjang kode untuk koneksi database, GUI, keamanan, dan validasi form. Hal ini memberikan manfaat dengan tidak perlu memikirkan beberapa kode atau fungsi dasar pada pembuatan aplikasi website, bahkan dalam penulisan baris kodenya bisa jauh lebih sedikit dari pada umumnya. (Wardana, 2010)

2.9.1 MVC

Model View Controller (MVC), adalah sebuah pola perancangan sistem pada perangkat lunak, dimana aplikasi dibagi menjadi 3 bagian, diantaranya (Basuki, 2014) :

1. *Model*, mempresentasikan data yang digunakan oleh aplikasi, contohnya database, RSS, data yang dipanggil oleh aplikasi dan melibatkan operasi create, read, update, dan delete.
2. *View*, informasi yang ditampilkan pada user melalui *browser*, biasanya dalam bentuk file HTML atau PHP yang menyusun sebuah template web.
3. *Controller*, merupakan jembatan antara *Model* dan *View*, dimana Controller akan merespon HTTP Request yang datang dari user, kemudian dari request memanggil fungsi pada model untuk mengolah data, diteruskan pada view supaya hasilnya dapat ditampilkan pada *browser*

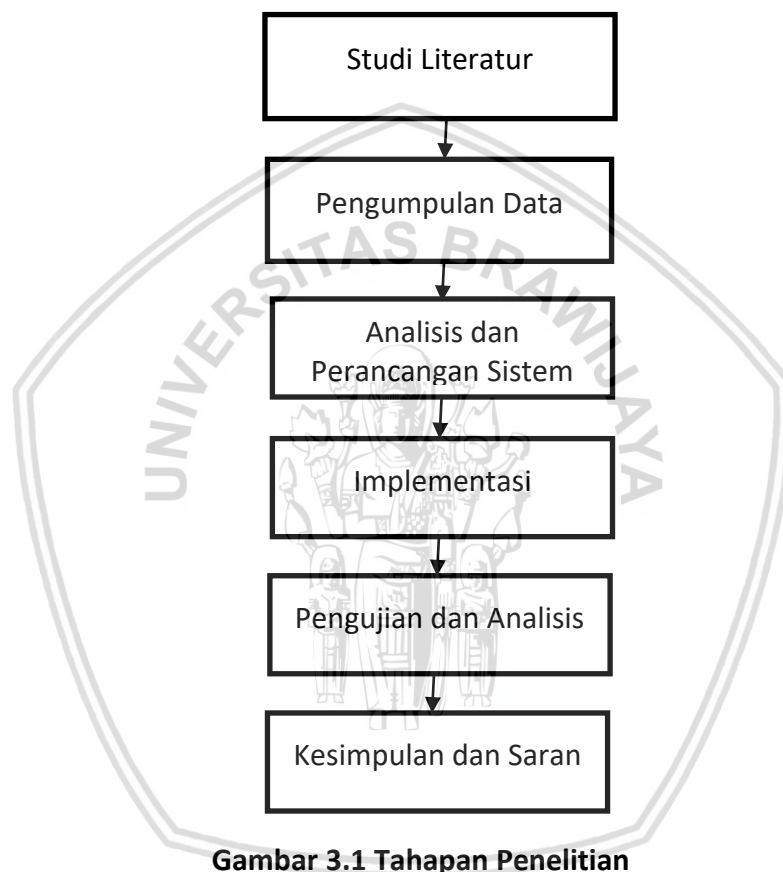
2.9.2 PHP

PHP adalah singkatan dari *Hypertext Preprocessor*, PHP merupakan bahasa pemrograman yang diproses di dalam server, dan hasilnya dikirimkan pada klien melalui browser. (Kadir, 2001). PHP secara khusus digunakan untuk membangun web yang dinamis, sehingga dapat membuat sesuai (Kadir, 2001).

PHP di buat oleh Rasmus Lerdorf, bahasa pemrograman web bekerja dalam webserver. Skrip-skrip PHP harus disimpan di dalam server dan dieksekusi dalam server tersebut sehingga web lebih interaktif dan dinamis (Madcoms, 2010).

BAB 3 METODOLOGI

Bab ini menjelaskan terkait tahapan yang akan dilakukan pada penelitian ini, diantaranya studi literatur, pengumpulan data, analisis dan perancangan sistem, implementasi, pengujian dan analisis, kesimpulan dan saran. Bagian kesimpulan dan saran diikutsertakan sebagai catatan untuk kemungkinan lebih lanjut dalam pengembangan aplikasi selanjutnya. Berikut alur tahapan dari metodologi dapat dilihat pada Gambar 3.1 berikut ini.



Gambar 3.1 Tahapan Penelitian

3.1 Studi Literatur

Dari penjelasan di atas studi literatur merupakan tahapan paling awal untuk melakukan penelitian ini, dikarenakan studi literatur digunakan untuk memahami konsep dan dasar teori yang akan diterapkan dalam penelitian “Penerapan Analisis Sentimen untuk Menilai Suatu Produk pada *Twitter* Berbahasa Indonesia dengan Metode *Naïve Bayes Classifier* dan Informaton Gain”. Berikut teori yang dipelajari untuk menunjang penelitian ini :

1. Text Mining
2. Analisis Sentimen

3. Metode *Naïve Bayes Classifier*
4. Metode *Information Gain*
5. Twitter

3.2 Pengumpulan Data

Tahapan ini bertujuan untuk mengumpulkan data twitter. Data yang diambil merupakan data primer melalui mekanisme pengambilan terstruktur tweet dari beberapa akun Twitter dengan memanfaatkan API twitter yang sudah disediakan khusus untuk para pengembang aplikasi, dan data yang diambil merupakan *tweet* yang berbahasa indoneisa serta dalam pencariannya menggunakan kata kunci nama produk tertentu, yaitu Mustika Ratu. Untuk menambahkan data yang lebih banyak lagi, peneliti mengadakan pengisian fomulir online untuk menuliskan sentimen terhadap produk Mustika Ratu oleh rekan-rekan peneliti.

Secara keseluruhan data yang diperoleh sebanyak 200 dokumen tweet, 100 untuk data yang berkategori positif, 100 untuk data yang berkategori negatif. Penetapan kelas atau kategori daripada 200 data tersebut, dilakukan dengan cara pengecekan dan validasi oleh ahli pakar yang memahami teori tentang sentimen.

3.3 Analisis Kebutuhan

Analisis kebutuhan dilakukan dalam menentukan kebutuhan-kebutuhan terkait dalam proses pembuatan sistem yang menerapkan analisis sentimen dengan menggunakan metode *Naïve Bayes* dan *Information Gain*. Berikut kebutuhan-kebutuhan yang diperlukan :

3.3.1 Kebutuhan Fungsional

Kebutuhan fungsional ada untuk menjelaskan hal-hal apa saja yang diberikan sistem untuk penggunaanya.

1. Sistem dapat mengubah dokumen tweet menjadi term term.
2. Sistem dapat mengganti kata alay yang ditemukan pada dokumen *tweet* menjadi kata baku sesuai dengan pengetahuan yang didapat oleh sistem pada database, fitur ini dinamakan normalisasi bahasa.
3. Sistem dapat mengklasifikasi kan dokumen pada kategori positif dan kategori negatif dengan menghitung tiap term dan mencari nilai probabilitasnya .

3.3.2 Kebutuhan Non Fungsional

1. Kebutuhan perangkat keras yang digunakan penulis :
 - a. Intel® Core™ i5-5200U CPU @ 2.20GHz 2.20 GHz
 - b. RAM 4 GB
 - c. Harddisk berkapasitas 600 GB

- d. Monitor 14 inc
- 2. Kebutuhan perangkat lunak :
 - a. Sistem Operasi Windows 10 Home
 - b. Xampp
 - c. Sublime 12.0
- 3. Data yang diperlukan :
 - a. Data Sentimen pada twitter (Komentar mengenai produk kecantikan Mustika Ratu).

3.4 Perancangan Sistem

Perancangan sistem dilakukan setelah pengumpulan data dan analisis kebutuhan sudah didapatkan, lebih lengkapnya seperti dibawah ini :

1. Dokumen *tweet* didapatkan dan sudah divalidasi oleh ahli pakar jumlah dokumen yang berkategori positif dan yang berkategori negatif.
2. Sebelum diklasifikasi dokumen *tweet* akan dilakukan proses text preprocessing dengan case folding, tokenizing, filtering, dan stemming, serta normalisasi bahasa untuk mendapatkan kata-kata unik, baku dan bersih dari karakter yang tidak dibutuhkan yang berpotensi mengganggu akurasi klasifikasi.
3. Setelah poses text preprocessing dilakukan, tahap selanjutnya dilakukan seleksi fitur dengan metode information gainnya, proses seleksi fitur ini digunakan untuk mengurangi keberadaan kata yang kurang relevan dan mengganggu proses klasifikasi.
4. Tahap selanjutnya dilakukan pencarian nilai likelihood pada tiap-tiap kata unik yang didapat sebelumnya dan mencari nilai prior dari jumlah dokumen yang ada, kemudian dicari nilai probabilitas positif dan probabilitas negatif dari tiap-tiap dokumen.
5. Penentuan kategori tiap dokumen berdasarkan nilai terbesar dari probabilitas positif maupun probabilitas negatif.
6. Setelah ditentukan kategori dari tiap dokumen, kemudian dicari nilai akurasinya oleh user secara manual.

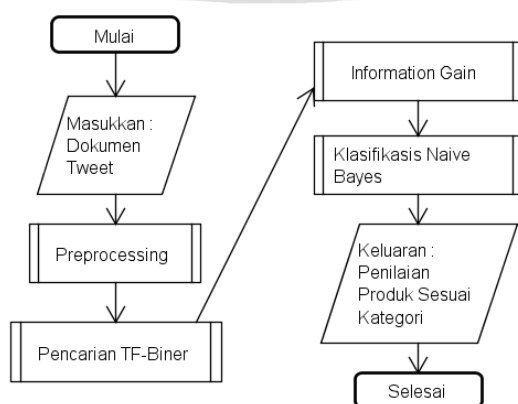
BAB 4 ANALISIS DAN PERANCANGAN

4.1 Deskripsi Umum Sistem

Sistem yang akan dibuat dalam penelitian ini adalah sebuah perangkat lunak yang dibangun dengan mengimplementasikan metode *Naïve Bayes Classifier* yang dikombinasikan dengan Metode *Information Gain* untuk analisis sentimen twit berbahasa Indonesia untuk menilai suatu produk. Data yang akan diolah adalah data twitter yang ada hubungannya dengan sentimen terhadap produk-produk kecantikan dari merek Mustika Ratu. Kemudian data-data tersebut akan diklasifikasi menjadi 2 bagian sentimen positif dan sentimen negatif. Penilaian produk berdasarkan nilai probabilitas positif maupun probabilitas negatif, semakin besar nilai probabilitas positif nya maka produk memiliki nilai kecenderungan pasar yang positif, begitu sebaliknya. Sistem ini akan menguji keakuratan hasil pengklasifikasian terhadap data twit tersebut dengan metode *Naïve Bayes* dan memanfaatkan metode *Information Gain* sebagai pemilihan fiturnya.

4.2 Perancangan Sistem

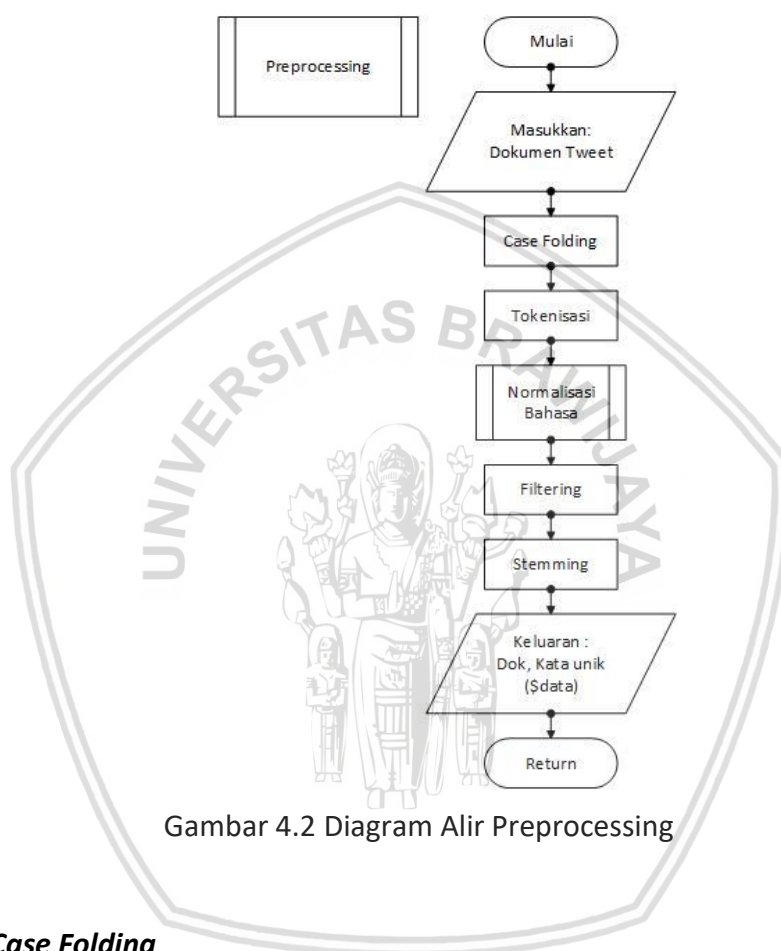
Untuk membangun sistem yang dapat digunakan untuk sentiment analisis dengan mengimplementasikan metode *Naïve Bayes Classifier* dan metode *Information Gain* sebagai pemilihan fiturnya dibuatlah sebuah diagram alir untuk membantu dalam proses memahami alur dari proses sistem berlangsung. Diagram alir tersebut menggambarkan tahapan-tahapan dalam menjalankan sistem, dimulai dari tahap memasukkan dokumen twit, kemudian dokumen tersebut memasuki proses preprocessing dan dirubah dalam bentuk vector term presence, setelah itu dilakukan pemilihan fitur dengan metode *Information Gain* nilai setiap fitur pada setiap dokumen, proses klasifikasi dokumen dengan *Naïve Bayes Classifier* dan menampilkan hasil klasifikasi, dan tahapan akhir menampilkan hasil dari proses klasifikasi hingga menghasilkan output penilaian produk. Pada Gambar 4.1 menunjukkan diagram alir sistem.



Gambar 4.1 Diagram Alir Sistem

4.2.1 Preprocessing

Dalam Proses Preprocessing terdapat beberapa langkah, yang pertama *case folding*, *tokenizing* dan normalisasi bahasa, kedua *filtering*, dan yang ketiga adalah *stemming*. Namun sebelum masuk tahap *filtering* dilakukan proses pengubahan kata Bahasa alay menjadi kata yang baku. Berikut Gambar 4.2 menunjukkan diagram alir dari proses *preprocessing*.



Gambar 4.2 Diagram Alir Preprocessing

4.2.1.1 Case Folding

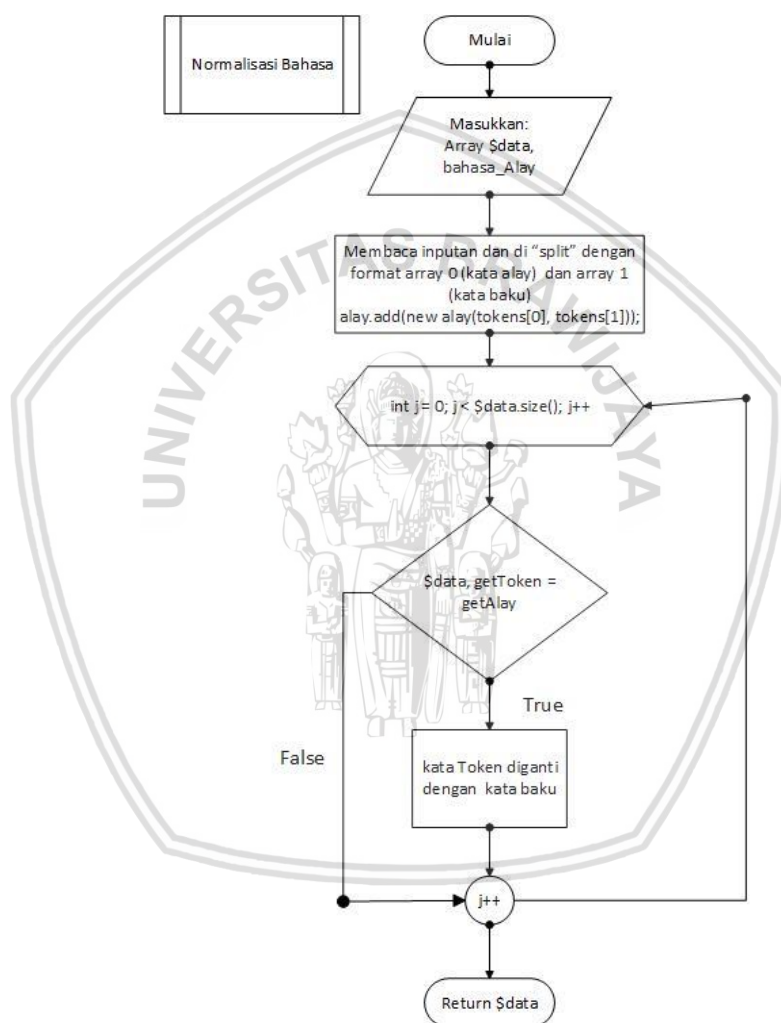
Proses ini adalah untuk merubah seluruh kata yang besar (Uppercase) menjadi huruf kecil (Lowercase). Data twitter yang sudah dikumpulkan oleh sistem akan berubah menjadi huruf kecil semua, hal ini tidak berlaku pada karakter selain huruf (angka, karakter, dan lain sebagainya).

4.2.1.2 Tokenizing

Pada proses *tokenizing* adalah mengubah dokumen menjadi kata dasar atau term dan mengubah semua huruf besar menjadi kecil. Kemudian penghapusan karakter selain huruf, seperti pada karakteristik postingan twit terdapat *noise* yang ada seperti URL, Hastag(#), dan Username (@) akan dihilangkan.

4.2.1.3 Normalisasi Bahasa

Proses normalisasi Bahasa adalah proses mengubah bahasa *alay* atau tidak baku menjadi bahasa yang baku sesuai Kamus Besar Bahasa Indonesia (KBBI). Proses ini dilakukan dengan cara mengganti kata *alay* yang dideteksi sebelumnya dengan menggunakan kamus bahasa alay, kemudian digantikan dengan kata yang sesuai KBBI. Setelah itu di simpan kembali untuk diproses lebih lanjut pada tahap berikutnya. Berikut Gambar 4.3 menunjukkan proses dari normalisasi bahasa :



Gambar 4.3 Diagram Alir Normalisasi Bahasa

4.2.1.4 Filtering

Proses Filtering adalah proses untuk menghilangkan kata-kata yang tidak penting yang kemudian disebut Stopword. Stopword merupakan kata yang sering muncul pada dokumen, yang tidak memiliki pengaruh dalam proses analisis sentimen, bahkan dapat menyebabkan banyaknya noise. Cara kerja

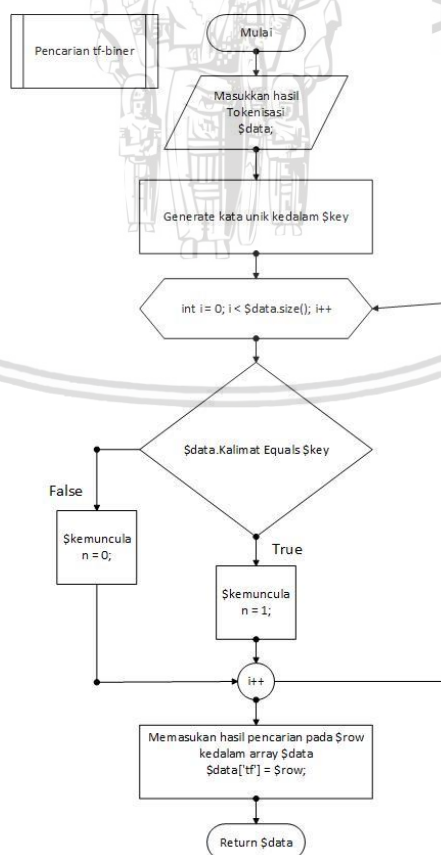
proses ini akan membandingkan kata – kata yang memiliki bentuk dan ciri serupa dengan Stopword yang sebelumnya dikumpulkan dalam satu file. Jika perbandingan kata tersebut memiliki nilai yang sama, maka akan dihilangkan kata yang ada didokumen.

4.2.1.5 Steaming

Proses Steaming adalah proses untuk menghilangkan imbuhan pada sebuah kata untuk merubah kata menjadi kata dasarnya. Penelitian ini menggunakan algoritma porter yang sudah disesuaikan untuk menjalankan proses steaming pada kata berbahasa Indonesia.

4.2.2 TF Biner

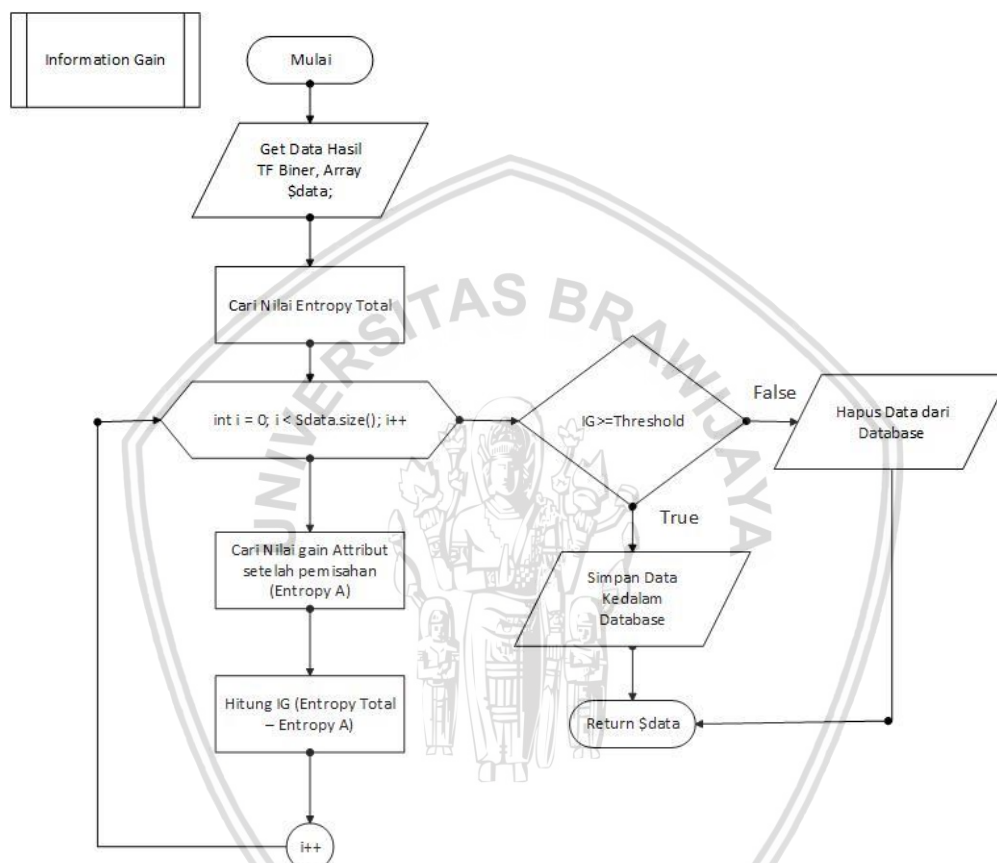
TF Biner ini digunakan untuk mencari keberadaan sebuah term pada dokumen. Caranya dengan merubah dokumen menjadi bentuk vektornya dengan diwakili oleh angka 0 dan angka 1. Angka 0 menunjukkan term atau token tidak terdapat pada dokumen tertentu, sedangkan angka 1 menunjukkan term atau token terdapat pada dokumen tertentu. Hasil dari proses TF-Biner ini akan diproses lagi kedalam proses seleksi. Gambar 4.4 berikut memaparkan proses dari TF-Biner.



Gambar 4.4 Diagram Alir TF-Biner

4.2.3 Proses Information Gain

Bagian tahap ini menjelaskan alir dari proses *Information Gain* sebagai tahap seleksi fitur. Data yang sudah diproses sebelumnya akan dicari nilai *Information Gain* pada tiap fitur (kata) yang ada pada data / dokumen. Fitur / kata yang memiliki nilai *Information Gain* dibawah ambang batas (threshold = 0,4) maka akan dihapus dari database, sedangkan nilai diatas maupun sama dengan nilai ambang batas akan tetap disimpan dalam tabel database. Gambar 4.5 menunjukkan proses diagram *Information Gain*.

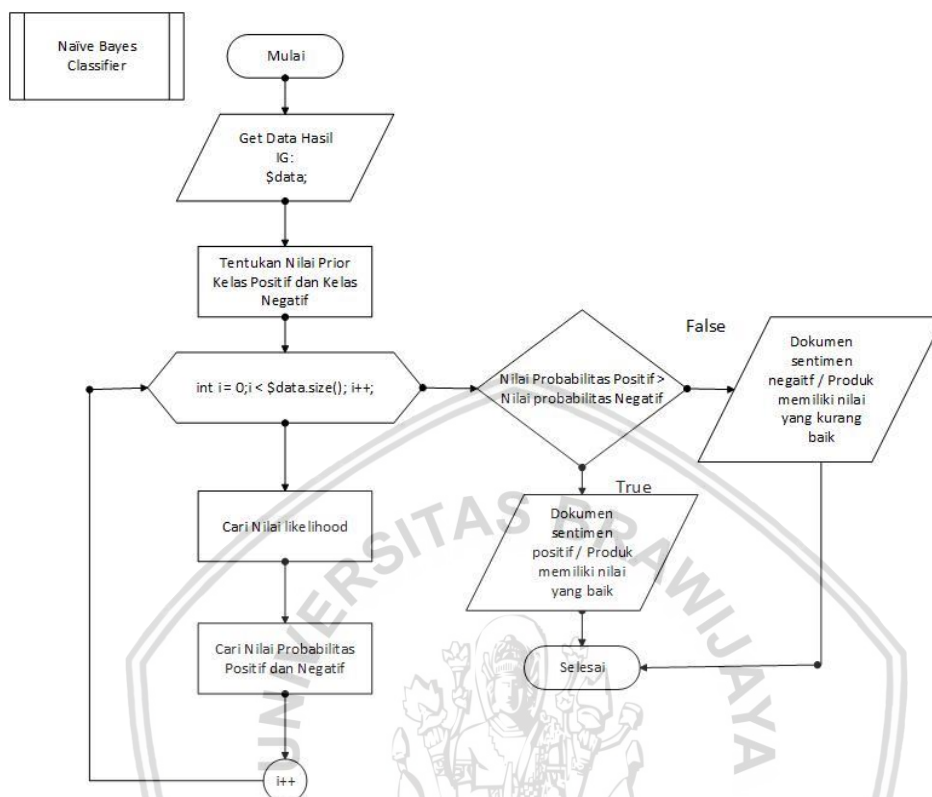


Gambar 4.5 Diagram Alir *Information Gain* (Feature Selection)

4.2.4 Proses Naive Bayes Classifier

Metode *Naive Bayes* merupakan metode klasifikasi yang paling sederhana dan mudah untuk diimplementasikan. Proses diawali dengan menentukan nilai prior tiap kategori yang ada (positif dan negatif), setelah itu mencari nilai likelihood tiap kata / fitur dalam database, kemudian mencari nilai probabilitas positif dari dokumen yang memiliki sejumlah likelihood sesuai dengan banyak kata yang ada, begitu pula nilai probabilitas negatif dari dokumen tersebut. Proses terakhir dengan membandingkan nilai probabilitas positif dan probabilitas negatif sebuah dokumen, jika nilai probabilitas positif lebih besar dari probabilitas negatif, maka sebuah dokumen akan dikategorikan sebagai

sentimen positif, atau dengan kata lain menambah nilai citra positif terhadap sebuah produk, begitu pula sebaliknya. Gambar 4.6 menunjukkan proses diagram alir dari *Naive Bayes Classifier* :



Gambar 4.6 Diagram Alir *Naive Bayes Classifier*

4.3 Manualisasi Perhitungan

Bagian sub bab ini menjelaskan pengolahan dataset yang sudah disediakan untuk diklasifikasikan menurut kelasnya. Adapun beberapa tahapan dimulai dari Perhitungan dataset, kemudian perhitungan Kemunculan Term pada Setiap Dokumen, *Feature selection Information Gain*, Pemobotan TF-IDF (WTD), hingga Klasifikasi *Naive Bayes*.

4.3.1 Dataset

Tabel 4.1 menunjukkan beberapa data/dokumen yang digunakan dalam penelitian, data/dokumen tersebut diambil pada akun twitter milik netizen yang membahas dalam ruang lingkup produk mustika ratu. Proses pengambilan data/dokumen tersebut melalui proses catching data/dokumen yang disediakan oleh Twitter API. Untuk pengfilteran data/dokumen menggunakan kata kunci mustika ratu baik berupa hastag maupun mention.

Tabel 4.1 Dataset Perhitungan Manualisasi

Dok	Dokumen	Kelas
Doc 1	Paket hemat biar kulit sehat #mustikaratu #maskerjafra https://www.instagram.com/p/BPj30khlw_/	Positif
Doc 2	Biarpun harganya murah tapi wanginya bikin adem apalagi yang pake cocok #mustikaratu #putri... https://www.instagram.com/p/BPbXbKtAAfU/	Positif
Doc 3	Cantik dan sehat itu penting @mustikaratuind #mustikaratu #hijau #girl https://www.instagram.com/p/BOHGg3Ng4ig/	Positif
Doc 4	Suka banget sama sabun wajah ini #mustikaratu #bengkoangwhitening Gak khawatir dech kalo... https://www.instagram.com/p/BlbfsCzDJx6/	Positif
Doc 5	Belajar make up alhasil lumayanlah,,, terimakasih mustika ratu	Positif
Doc 6	Perempuan jangan mau diPHPin #salam2jari#tetapahokdjarot#IniBaruJakarta#mustikaratu	Negatif
Doc 7	Mustikaratu terlalu keras buat kulit sensitif. Pilih yang sesuai kebutuhan kulit kamu. Jangan pake karna ikut ikutan orang lain.	Negatif
Doc 8	Campur mengkudu bg "@Gondes696: Padahal udah pake mustikaratu sari bengkoang #huft"@joehanbyd: Karna abg jelek haha "@Gondes696: Kenapa ak	Negatif
Doc 9	mustika ratu udh jelek produknya, skng mah tje fuk yg bisa bikin buka kyk hantu beneran RT @reyreynatha: Aduh, harusnya mustika ratu bkin	Negatif
Doc 10	Tadi terrbeli rose water mustikaratu,adik cakap seram laa bau dia,rose water tu kan org guna waktu kapan jenazah.pastu aku cakap	Negatif
Doc 11	Pulang ngampus langsung bersihin muka pake mustika ratu si mama :D yeeayy anak yg baik :) [pic] — https://path.com/p/ycQ2E	Positif
Doc 12	Aku suka banget pakai salah satu rangkaian produk Mustika Ratu Whitening Series, salah satu... https://www.instagram.com/p/BMDJmfVj1kL/	Positif
Doc 13	Salah beli lulur -_- lulur kocok yg coklat itu kok skrg uda ga ada dipasaran lg yah? Pdhl itu dr MustikaRatu loh. Yg coklat jelek itu	Negatif
Doc 14	Mustika ratu kurang cocok kalu buat muka aku	Negatif
Doc 15	@Nanda_Wong yah ada lha nda dari mustika ratu, enak alami tanpa kimia mantab...("~")ε"	Positif

Setelah dataset dikumpulkan, terdapat beberapa proses sebelum dilakukan pengklasifikasian data/dokumen, salah satunya adalah tahapan preprocessing, dokumen akan diproses melalui tahapan tokenisasi, normalisasi bahasa *alay*, filtering, dan stemming. Pada Tabel 4.2 dipaparkan hasil dari proses Preprocessing.

Tabel 4.2 Hasil Tokenisasi

Dok	Dokumen	Kelas
Doc 1	paket hemat biar kulit sehat	Positif
Doc 2	biarpun harganya murah tapi wanginya bikin adem apalagi yang pake cocok	Positif
Doc 3	cantik dan sehat itu penting	Positif
Doc 4	suka banget sama sabun wajah ini gak khawatir deh kalo	Positif
Doc 5	belajar make up alhasil lumayanlah terimakasih mustika ratu	Positif
Doc 6	perempuan jangan mau diPHPin	Negatif
Doc 7	mustikaratu terlalu keras buat kulit sensitif pilih yang sesuai kebutuhan kulit kamu jangan pake karna ikut ikutan orang lain.	Negatif
Doc 8	campur mengkudu bg padahal udah pake mustikaratu sari bengkoang karna abg jelek haha kenapa ak	Negatif
Doc 9	mustika ratu udh jelek produknya skng mah tje fuk yg bisa bikin buka kyk hantu beneran rt aduh harusnya mustika ratu bkin	Negatif
Doc 10	tadi terrbeli rose water mustikaratu adik cakap seram laa bau dia rose water tu kan org guna waktu kapan jenazah pastu aku cakap	Negatif
Doc 11	pulang kampus langsung bersihin muka pakai mustika ratu si mama yeah anak yg baik	Positif
Doc 12	<u>aku suka banget pakai salah satu rangkaian produk mustika ratu whitening series salah satu</u>	Positif
Doc 13	salah beli lulur lulur kocok yg coklat itu kok skrg uda ga ada dipasaran lg yah pdhl itu dr mustika ratu loh yg coklat jelek itu	Negatif
Doc 14	mustika ratu kurang cocok kalo buat muka aku	Negatif
Doc 15	yah ada lha nda dari mustika ratu enak alami tanpa kimia mantab	Positif

Setelah proses Tokenisasi, pada Tabel 4.3 menunjukkan hasil dari proses normalisasi bahasa alay, merubah bahasa alay menjadi bahasa indonesia yang baku.

Tabel 4.3 Hasil Normalisasi Bahasa

Dok	Dokumen	Kelas
Doc 1	paket hemat biar kulit sehat	Positif
Doc 2	biarpun harganya murah tapi wanginya buat adem apalagi yang pakai cocok	Positif
Doc 3	cantik dan sehat itu penting	Positif
Doc 4	suka banget sama sabun wajah ini tidak khawatir deh kalau	Positif
Doc 5	belajar make up alhasil lumayan lah terima kasih mustika ratu	Positif
Doc 6	perempuan jangan mau di tipu	Negatif
Doc 7	mustikaratu terlalu keras buat kulit sensitif pilih yang sesuai kebutuhan kulit kamu jangan pakai karena ikut ikutan orang lain.	Negatif
Doc 8	campur mengkudu abang padahal sudah pakai mustikaratu sari bengkoang karena abang jelek haha kenapa aku	Negatif
Doc 9	mustika ratu sudah jelek produknya sekarang mah tje fuk yang bisa buat buka kayak hantu bener rt aduh harusnya mustika ratu buat	Negatif
Doc 10	tadi beli rose water mustikaratu adik cakap seram lah bau dia rose water itu kan orang guna waktu kapan jenazah pasti aku cakap	Negatif
Doc 11	pulang kampus langsung bersihin wajah pakai mustika ratu si mama yeah anak yg baik	Positif
Doc 12	<u>aku suka banget pakai salah satu rangkaian produk mustika ratu whitening series salah satu</u>	Positif
Doc 13	salah beli lulur lulur kocok yang coklat itu kok sekarang sudah tidak ada dipasaran lagi yah padahal itu dari mustika ratu loh yang coklat jelek itu	Negatif
Doc 14	mustika ratu kurang cocok kalau buat wajah aku	Negatif
Doc 15	yah ada lah dari mustika ratu enak alami tanpa kimia mantab	Positif

Tabel 4.4 merupakan hasil akhir dari tahap preprocessing dimana melibatkan proses filtering dan proses stemming.

Tabel 4.4 Hasil Akhir *Preprocessing*

Dok	Dokumen	Kelas
Doc 1	paket hemat biar kulit sehat	Positif
Doc 2	biar harga murah wangi buat adem pakai cocok	Positif
Doc 3	cantik sehat penting	Positif
Doc 4	suka banget sabun wajah tidak khawatir	Positif
Doc 5	belajar make up alhasil lumayan terima kasih	Positif
Doc 6	perempuan jangan mau tipu	Negatif
Doc 7	terlalu keras buat kulit sensitif pilih sesuai kebutuhan kulit kamu jangan pakai ikut ikut orang.	Negatif
Doc 8	campur mengkudu abang pakai sari bengkoang abang jelek kenapa aku	Negatif
Doc 9	jelek produk bisa buat buka kayak hantu bener buat	Negatif
Doc 10	beli rose water adik cakap seram bau dia rose water orang guna waktu jenazah pasti aku cakap	Negatif
Doc 11	pulang kampus langsung bersih wajah pakai mama anak baik	Positif
Doc 12	aku suka banget pakai rangkaian produk whitening series	Positif
Doc 13	salah beli lulur lulur kocok coklat tidak ada pasar lagi coklat jelek	Negatif
Doc 14	kurang cocok buat wajah aku	Negatif
Doc 15	ada enak alami tanpa kimia mantab	Positif

4.3.2 Menghitung TF-Biner

Dokumen yang sudah melewati tahap Preprocessing masih berupa dalam bentuk string selanjutnya diolah menjadi bentuk vektornya, proses ini dilakukan guna mencari nilai TF Biner pada setiap dokumen. Proses Perhitungan TF-Biner dilakukan dengan memberikan nilai 1 dan 0, dimana nilai 1 akan diberikan ketika term tersebut muncul dalam dokumen, sedangkan nilai 0 diberikan ketika term tidak muncul dalam dokumen. Berikut dipaparkan perhitungan Term Presence pada Tabel 4.5 :

Tabel 4.5 Manualisasi Perhitungan TF-Biner

Dok	paket	Hemat	biar	Kulit	sehat	Murah	Seram	bau	Produk	guna	jenazah	tidak	salah
Doc 1	1	1	1	1	1	0	0	0	0	0	0	0	0
Doc 2	0	0	1	0	0	1	0	0	0	0	0	0	0
Doc 3	0	0	0	0	1	0	0	0	0	0	0	0	0
Doc 4	0	0	0	0	0	0	0	0	0	0	0	1	0
Doc 5	0	0	0	0	0	0	0	0	0	0	0	0	0
Doc 6	0	0	0	0	0	0	0	0	0	0	0	0	0
Doc 7	0	0	0	0	0	0	0	0	0	0	0	0	0
Doc 8	0	0	0	0	0	0	0	0	1	0	0	0	0
Doc 9	0	0	0	0	0	0	0	0	0	0	0	0	0
Doc 10	0	0	0	0	0	0	1	1	0	1	1	0	0
Doc 11	0	0	0	0	0	0	0	0	0	0	0	0	0
Doc 12	0	0	0	0	0	0	0	0	1	0	0	0	0
Doc 13	0	0	0	0	0	0	0	0	0	0	0	1	1
Doc 14	0	0	0	0	0	0	0	0	0	0	0	0	0
Doc 15	0	0	0	0	0	0	0	0	0	0	0	0	0

4.3.3 Manualisasi Information Gain

Setelah dilakukan pembobotan TF-IDF pada tiap dokumen selanjutnya dilakukan penyeleksian fitur dari setiap dokumen, untuk mengurangi dimensi yang terlalu melebar. Adapun beberapa tahapan yang harus dilakukan :

1. Inisialisasi Parameter yang digunakan, antara lain :
 - a. Threshold = 0.07
 - b. Data Sentimen Positif : 8
 - c. Data Sentimen Negatif : 8

2. Menghitung nilai Entropy Total (Info(D))

Nilai Entropy Total didapatkan sesuai dengan persamaan 2.5 berupa penjumlahan Entropy pada kelas positif dan Entropy pada kelas negatif sesuai jumlah kelas yang ada ($i = 1, 2, \dots, c$) dimana c adalah banyak kelas / jumlah klasifikasi. Entropy didapatkan dari perkalian proporsi jumlah sampel kategori i dengan sampel total dengan log berbasis 2 terhadap proporsi jumlah sampel kategori i dengan sampel total. Hasil perhitungan Entropy Total dapat dilihat pada Tabel 4.6 berikut :

Tabel 4.6 Perhitungan Entropy Total

Entropy Total	Kelas Positif	
Info(D)	$-(8/16 * (\log_2(8/16;2)))$	0,5
	$-(8/16 * (\log_2(8/16;2)))$	0,5
	Total	1

3. Menghitung nilai Entropy tiap fitur setelah pemisahan / splitting
- Pemisahan yang dimaksudkan dalam manualisasi ini berdasarkan jenis nilai yg dimiliki oleh fitur yang berupa kata pada tiap dokumen ($j = 1, 2, \dots, v$). Pada tabel TF Biner 4.5, nilai tiap fitur adalah 0 (tidak ada) dan 1 (ada), sehingga jumlah pemisahan sesuai dengan persamaan 3.8 dimana membandingkan jumlah sample pada partisi j dengan jumlah keseluruhan sample, kemudian dikali dengan nilai entropy setiap partisi j . Untuk lebih jelasnya dapat dilihat pada Tabel berikut :

Tabel 4.7 Perhitungan Entropy fitur “biar” setelah pemisahan

1	Entropy Partisi (1) ada =	$-\sum_{i=1}^c P(i) \log_2 P(i)$	$-\sum_{i=1}^2 \frac{2}{2} \log_2 \frac{2}{2}$	= 0
		$-\sum_{i=2}^c P(i) \log_2 P(i)$	$-\sum_{i=2}^2 \frac{0}{2} \log_2 \frac{0}{2}$	= 0
	Jumlah	Total		= 0
2	Entropy Partisi (0) tidak ada =	$-\sum_{i=1}^c P(i) \log_2 P(i)$	$-\sum_{i=1}^2 \frac{6}{14} \log_2 \frac{6}{14}$	= 0,5239
		$-\sum_{i=2}^c P(i) \log_2 P(i)$	$-\sum_{i=2}^2 \frac{8}{14} \log_2 \frac{8}{14}$	= 0,4613
	Jumlah	Total		= 0,9852
3	Hitung Nilai Entropy Fitur setelah	$-\sum_{j=1}^v \frac{ D_j }{ D } \text{Info} D_j$	$-\sum_{j=1}^1 \frac{2}{16} \times 0$	

	pemisahan		
	(InfoA(D)) =	$-\sum_{j=1}^2 \frac{ D_j }{ D } \text{Info}D_j$	$-\sum_{j=1}^1 \frac{14}{16} \times 0,9852$
	Jumlah	Total	= 0,8620

4. Menghitung nilai *Information Gain* Fitur (InfoGain(A))

Tabel 4.8 Perhitungan *Information Gain* Fitur “biar”

No	Langkah – langkah		
1	Hitung Nilai Informatin Gain Fitur	$-\text{Info}(D) - \text{Info}_A(D) $	
2	InfoGain(A) =	$1 - 0,8620$	= 0,1379

Berikut Tabel 4.9 menjelaskan hasil keseluruhan nilai *Information Gain* dari tiap fitur / kata, bagian yang ditandai warna oranye akan di hapus dari database karna kurang dari threshold (0.07) :

Tabel 4.9 Hasil Keseluruhan Nilai InfoGain(A)

Fitur	infoGain(A)
paket	0,06550785
hemat	0,06550785
Biar	0,13792538
Kulit	0,06550785
sehat	0,13792538
harga	0,06550785
orang	0,13792538
Jelek	0,21899526
Hantu	0,06550785
Beli	0,13792538

4.3.4 Klasifikasi Dokumen dengan Naive Bayes Classifier

Tahapan Klasifikasi dapat dilakukan setelah melewati beberapa tahap seleksi fitur diatas. Mulanya akan dicari nilai Prior dari masing – masing kelas yang ada ($p(C_j)$, dimana $j = 1,2,3,...,k$). Setelah itu dicari nilai likelihood dari setiap kata yang ada dalam database. Sehingga dapat diketahui nilai Posterior dokumen. Kemudian akan dibandingkan nilai Posterior kelas positif dengan Posterior kelas negatif, jika lebih besar Posterior kelas positif dari pada Posterior kelas negatif, maka dokumen termasuk dalam kategori kelas positif, begitu sebaliknya. Adapun tahapan sebelum dilakukan perhitungan klasifikasi *Naive Bayes*, diantaranya :

1. Inisialisasi jumlah kata unique, kata dalam kelas positif, dan kata dalam kelas negatif :
 - a. Jumlah kata unique pada data training = 95
 - b. Jumlah kata dalam kelas positif = 48
 - c. Jumlah kata dalam kelas negatif = 66
2. Hitung nilai Prior :

Tabel 4.10 Langkah Perhitungan Prior

No	Langkah – langkah		
1	Hitung Prior Kelas Positif =	Jumlah dokumen positif / Keseluruhan dokumen	$8/16 = 0,5$
2	Hitung Prior Kelas Negatif =	Jumlah dokumen negatif / Keseluruhan dokumen	$8/16 = 0,5$

3. Hitung nilai likelihood tiap kata :
 Dalam Manualisasi ini akan menghitung nilai likelihood kata “biar” :

Tabel 4.11 Langkah Perhitungan Likelihood kata “biar”

No	Langkah – langkah		
1	Hitung Likelihood =	$(2+1)/(48+95)$	= 0,020979
		2 didapat dari banyak kata “biar” yang ada dalam dokumen, sedangkan angka + 1	

		diberikan untuk menghindari nilai 0, teknik ini dikenal dengan sebutan "Laplace Smoothing"	
--	--	--	--

Berikut Tabel 4.12 menjelaskan hasil keseluruhan nilai likelihood dari tiap fitur / kata :

Tabel 4.12 Hasil Keseluruhan Perhitungan Likelihood tiap fitur/kata

Fitur	Likelihood	
	Kelas Positif	Kelas Negatif
paket	0,006993007	0,006369427
Hemat	0,006993007	0,006369427
Biar	0,020979021	0,006369427
kulit	0,006993007	0,006369427
sehat	0,020979021	0,006369427
harga	0,006993007	0,006369427
orang	0,006993007	0,01910828
jelek	0,006993007	0,025477707
hantu	0,006993007	0,006369427
beli	0,006993007	0,01910828

4. Mengklasifikasikan beberapa dokumen untuk mencari nilai posterior dari dokumen tersebut untuk masuk kedalam kelas positif atau negatif. Berikut dipaparkan hasil dari perhitungan terhadap data testing pada Tabel 4.13 :

Tabel 4.13 Hasil Klasifikasi *Naive Bayes* terhadap Dokumen 1

DOKUMEN 1			KELAS
Suka banget sabun wajah tidak khawatir			POSITIF
Posterior Kelas Positif		Posterior Kelas Negatif	
Positif	0,5	negatif	0,5
Suka	0,020979021	suka	0,00621118

Banget	0,020979021	banget	0,00621118
Sabun	0,006993007	sabun	0,00621118
Wajah	0,006993007	Wajah	0,00621118
Tidak	0,006993007	Tidak	0,00621118
Khawatir	0,006993007	khawatir	0,00621118
TOTAL	5,26255E-13		2,8709E-14

Tabel 4.14 Hasil Klasifikasi *Naive Bayes* terhadap Dokumen 2

DOKUMEN 2				KELAS
kurang cocok buat wajah aku				NEGATIF
Posterior Kelas Positif		Posterior Kelas Negatif		
Positif	0,5	Negatif	0,5	
kurang	0,006993007	Kurang	0,00621118	
Cocok	0,006993007	Cocok	0,00621118	
Buat	0,006993007	Buat	0,00621118	
wajah	0,006993007	wajah	0,00621118	
Aku	0,013986014	Aku	0,03184713	
TOTAL	1,67232E-11			2,3699E-11

Tabel 4.15 Hasil Klasifikasi *Naive Bayes* terhadap Dokumen 3

DOKUMEN 3				KELAS
jelek produk bisa buat buka kayak hantu bener buat				NEGATIF
Posterior Kelas Positif		Posterior Kelas Negatif		
Positif	0,5	Negatif	0,5	
jelek	0,006993007	jelek	0,025477707	
produk	0,006993007	produk	0,00621118	
Bisa	0,006993007	Bisa	0,00621118	
Buat	0,006993007	Buat	0,00621118	
Buka	0,006993007	Buka	0,00621118	
Kayak	0,006993007	kayak	0,00621118	

Hantu	0,006993007	hantu	0,00621118
Bener	0,006993007	bener	0,00621118
	2,85944E-18		4,54307E-18

Tabel 4.16 Hasil Klasifikasi *Naive Bayes* terhadap Dokumen 4

DOKUMEN 4			KELAS
ada enak alami tanpa kimia mantab			POSITIF
Posterior Kelas Positif		Posterior Kelas Negatif	
Positif	0,5	negatif	0,5
Ada	0,006993007	Ada	0,00621118
Enak	0,006993007	enak	0,00621118
Alami	0,006993007	alami	0,00621118
Tanpa	0,006993007	tanpa	0,00621118
Kimia	0,006993007	kimia	0,00621118
mantab	0,006993007	mantab	0,00621118
	5,84728E-14		2,87088E-14

Tabel 4.17 Hasil Klasifikasi *Naive Bayes* terhadap Dokumen 5

DOKUMEN 5			KELAS
wajah makin jelek pakai			NEGATIF
Posterior Kelas Positif		Posterior Kelas Negatif	
Positif	0,5	negatif	0,5
Wajah	0,006993007	wajah	0,00621118
Makin	0,006993007	makin	0,00621118
Jelek	0,006993007	jelek	0,025477707
Pakai	0,006993007	pakai	0,00621118
	1,19571E-09		3,05248E-09

Tabel 4.18 Hasil Klasifikasi *Naive Bayes* terhadap Dokumen 6

DOKUMEN 6		KELAS	
seru suka banget sama produk soal selalu inspirasi kaya alam indonesia		POSITIF	
Posterior Kelas Positif		Posterior Kelas Negatif	
Positif	0,5	Negatif	0,5
Seru	0,006993007	Seru	0,00621118
Suka	0,020979021	Suka	0,00621118
Banget	0,020979021	banget	0,00621118
Sama	0,006993007	Sama	0,00621118
Produk	0,006993007	produk	0,00621118
Soal	0,006993007	Soal	0,00621118
Selalu	0,006993007	selalu	0,00621118
Inspirasi	0,006993007	inspirasi	0,00621118
Kaya	0,006993007	kaya	0,00621118
Alam	0,006993007	Alam	0,00621118
Indonesia	0,006993007	indonesia	0,00621118
TOTAL	1,79965E-19		6,87919E-21

Dari ke empat Tabel diatas, didapatkan hasil manualisasi klasifikasi pada Dokumen 1 posterior positif sebesar 5,26255E-13 dan posterior negatif sebesar 2,87088E-14, maka Dokumen 1 termasuk kategori kelas positif. Dokumen 2 posterior positif sebesar 1,67232E-11 dan posterior negatif sebesar 2,36994E-11, maka Dokumen 2 termasuk kategori kelas negatif. Dokumen 3 posterior positif sebesar 2,85944E-18 dan posterior negatif sebesar 4,54307E-18, maka Dokumen 3 termasuk kategori kelas negatif. Sedangkan Dokumen 4 posterior positif sebesar 5,84728E-14 dan posterior negatif sebesar 2,87088E-14, maka Dokumen 4 termasuk kategori positif. Kemudian Dokumen 5 posterior positif sebesar 1,19571E-09 dan posterior negatif 3,05248E-09, maka Dokumen 5 termasuk kategori negatif. Setelah itu untuk Dokumen 6 posterior positif sebesar 1,79965E-19, dan posterior negatif sebesar 6,87919E-21, maka termasuk kategori positif.

Sedangkan untuk perhitungan Akurasi atas hasil klasifikasi diatas, penelitian ini menggunakan Tabel *Matrix Confusion*, seperti dibawah ini :

Tabel 4.19 *Matrix Confusion* untuk mencari Nilai Akurasi

		Prediksi	
		Negatif	Positif
Actual	Negatif	A	B
	Positif	C	D

Keterangan :

- A, jumlah prediksi yang benar bahwa yang diprediksi bernilai negatif
- B, jumlah prediksi yang salah, yang seharusnya nilai negatif diprediksi positif
- C, jumlah prediksi yang salah, yang seharusnya nilai positif diprediksi negatif
- D, jumlah prediksi yang benar bahwa yang diprediksi positif

Akurasi adalah proporsi dari jumlah prediksi yang benar dari semua data yang diprediksi, yaitu dengan rumus :

$$\text{Akurasi Prediksi} : \frac{A+D}{A+B+C+D} \quad (3.1)$$

4.4 Perancangan Pengujian

Perancangan Pengujian pada skripsi ini berfokus untuk menunjukkan bahwa perangkat lunak telah mampu bekerja sesuai dengan spesifikasi kebutuhan yang ada, pengujian yang akan dilakukan dengan scenario mencari nilai akurasi pada data uji dengan menggunakan tabel *Matrix Confusion*. Nilai akurasi akan dicari dengan threshold yang beragam pada metode *Information Gain*-nya, dengan kisaran threshold 0 – 0.10. Saat nilai threshold = 0, maka diasumsikan metode *Information Gain* tidak digunakan untuk menfilter data latih, sedangkan saat threshold > 0, hingga threshold = 0.10 maka dapat disimpulkan seleksi fitur *Information Gain* diterapkan pada proses analisis sentimen. Kemudian saat nilai akurasinya sudah didapatkan, maka akan dicari nilai terbaiknya.

BAB 5 IMPLEMENTASI

5.1 Implementasi Program

Berdasarkan perancangan sistem yang ada pada bab 3, maka proses implementasi dijelaskan pada subbab berikut ini. Adapun daftar fungsi yang digunakan untuk memperjelas fungsi-fungsi yang ada pada kode program saat proses implementasi berlangsung.

Tabel 5.1 Daftar Fungsi

No.	Nama Fungsi	Kelas	Kegunaan
1	get_data_stem_stop()	Text_processing.php (Controller)	Untuk mendapatkan hasil tokenisasi
2	get_kata_unik()	Text_processing.php (Controller)	Menyimpan hasil tokenisasi (kata unik) kedalam array "words"
3	term_frequency()	Text_processing.php (Controller)	Mencari nilai tf biner padaa tiap-tiap dokumen
4	get_entropy()	Text_processing.php (Controller)	Mencari nilai entorpy total pada jumlah dokumen yang ada.
5	get_entropy_partisi_P_N()	Text_processing.php (Controller)	Mecari nilai entropy partisi ada dan partisi tidak ada pada seluruh dokumen
6.	info_gain()	Text_processing.php (Controller)	Untuk mencari nilai infogain pada tiap tiap fitur (kata)
7	frekuensi_dokumen2()	Text_processing.php (Controller)	Mencari nilai tf untuk kebutuhan

			pencarian likelihood nantinya.
8	stop_removal(\$data)	M_processing.php (Model)	Bertugas menghapus kata dan karakter yang tidak penting pada tiap dokumen
9	kata_alay(\$words)	M_processing.php (Model)	Mengganti kata alay yang terdeteksi pada dokumen dan diganti dengan kata baku sesuai kbki yang disimpan dalam database sebelumnya
10	get_data_training()	M_processing.php (Model)	Mendapatkan data training dari database
11	likelihood()	Naive_bayes.php (Controller)	Mencari nilai likelihood
12	simpan_likelihood()	Naive_bayes.php (Controller)	Menyimpan nilai likelihood tiap fitur / kata pada array dan diserahkan pada kelas Model
13	hapus_likelihood_get_prior()	Naive_bayes.php (Controller)	Menghapus tabel likelihood saat dilakukan generate pada program supaya dapat di isi dengan nilai likelihood yang baru, sekaligus

			mendapatkan nilai prior dan menyimpannya pada database
14	simpan_likelihood(\$data)	M_naivebayes.php (Model)	Menyimpan nilai likelihood pada database setelah di proses oleh kelas controller
15	simpan_prior(\$data)	M_naivebayes.php (Model)	Menyimpan nilai prior pada database setelah di proses oleh kelas controller
16	hapus_likelihood_get_prior(\$data)	M_naivebayes.php (Model)	Memasukkan data nilai prior pada tabel setelah penghapusan nilai likelihood
17	hapus_likelihood()	M_naivebayes.php (Model)	Menghapus likelihood pada database setelah di proses oleh kelas controller
18	kalimat_positif()	M_naivebayes.php (Model)	Menghitung jumlah dokumen positif pada database
19	kalimat_negatif()	M_naivebayes.php (Model)	Menghitung jumlah dokumen negatif pada database
20	testing()	Tester.php (Controller)	Melakukan preprocessing pada data uji

21	get_likelihood_dan_prior()	Tester.php (Controller)	Mendapatkan nilai likelihood dan nilai prior pada database
22	hitung_probablity()	Tester.php (Controller)	Menghitung nilai probabilitas pada data uji, dan menentukan kelas suatu dokumen yang merupakan data uji
23	get_data_uji()	Pengujian.php (Controller)	Mengambil data uji dari database tabel uji guna 10-fold cross validation
24	testing()	Pengujian.php (Controller)	Melakukan preprocessing pada data uji dari database tabel uji guna 10-fold cross validation
25	get_likelihood_dan_prior()	Pengujian.php (Controller)	Mendapatkan nilai likelihood dan nilai prior pada database tabel uji guna 10-fold cross validation
26	function hitung_probablity()	Pengujian.php (Controller)	Menghitung nilai probabilitas dan menentukan kelas suatu dokumen yang merupakan data uji pada database tabel

			uji, sekaligus menyimpan hasil klasifikasi pada tabel tester_hasil guna 10-fold cross validation
27	get_data_uji()	M_pengujian.php (Model)	Mengambil data uji dari database tabel uji setelah proses pada kelas controller
28	simpan(\$data)	M_pengujian.php (Model)	Menyimpan hasil klasifikasi pada tabel tester_hasil setelah proses pada kelas controller guna 10-fold cross validation

5.1.1 Load Dokumen

Pada bagian load dokumen, ditunjukkan bagian kode program yang memproses tahap meload dokumen pada tabel view dokumen, sesuai dengan konsep MVC yang sudah dijelaskan pada bab 2, maka penjelasan akan di jelaskan berdasarkan kelas Model, View, dan Controller.

Proses M_training_data.php adalah kelas model yang digunakan untuk melakukan proses pengambilan data di database

Kode Program 5.1 M_training_data.php

1	<?php if (! defined('BASEPATH')) exit('No direct script access allowed');
2	
3	class M_training_data extends CI_Model {
4	
5	
6	var \$table = 'data_training';
7	var \$column = array('kalimat','status'); //set column field database for order and search

8	var \$order = array('id' => 'desc');
9	
10	
11	//fungsi memanggil query untuk mengolah data latih dari database
12	private function _get_datatables_query()
13	{
14	
15	\$this->db->from(\$this->table);
16	
17	\$i = 0;
18	
19	foreach (\$this->column as \$item) // perulangan pada kolom
20	{
21	if(\$_POST['search']['value']) // jika datatable mengirimkan POST untuk memulai pencarian
22	{
23	
24	if(\$i===0) // first loop
25	{
26	\$this->db->group_start(); // open bracket. query Where with OR clause better with bracket. because maybe can combine with other WHERE with AND.
27	
28	
29	\$this->db->like(\$item, \$_POST['search']['value']);
30	}
31	Else
32	{
33	\$this->db->or_like(\$item, \$_POST['search']['value']);
34	
35	}
36	
37	if(count(\$this->column) - 1 == \$i) //last loop
38	\$this->db->group_end(); //close bracket

39	}
40	\$column[\$i] = \$item; // set column array variable to order processing
41	\$i++;
42	}
43	
44	if(isset(\$_POST['order'])) // here order processing
45	{
46	\$this->db->order_by(\$column[\$_POST['order']]['0']['column'], \$_POST['order']]['0']['dir']);
47	
48	}
49	else if(isset(\$this->order))
50	{
51	\$order = \$this->order;
52	
53	\$this->db->order_by(key(\$order), \$order[key(\$order)]);
54	}
55	}
56	
57	//fungsi untuk menghitung jumlah baris
58	public function count_filtered()
59	{
60	\$this->_get_datatables_query();
61	\$query = \$this->db->get();
62	
63	return \$query->num_rows();
64	}
65	public function count_all()
66	{
67	\$this->db->from(\$this->table);
68	
69	return \$this->db->count_all_results();
70	}
71	

72	//fungsi untuk menampilkan isi dari database dengan meneruskannya ke kelas controller kemudian ke kelas
73	public function tampil()
74	{
75	\$this->_get_datatables_query();
76	if(\$_POST['length'] != -1)
77	\$this->db->limit(\$_POST['length'], \$_POST['start']);
78	
79	\$query = \$this->db->get();
80	return \$query->result();
81	}
82	}

Sedangkan pada kelas controller, C_Training_data.php bertugas sebagai pihak yang membawa data yang sudah di simpan kelas M_training_data.php ke bagian kelas view untuk ditampilkan tabel dokumen twitter sesuai dengan isi yang ada pada database.

Kode Program 5.2 C_training_data.php

1	<?php
2	defined('BASEPATH') OR exit('No direct script access allowed');
3	
4	class C_Training_data extends CI_Controller {
5	
6	function __construct(){
7	parent::__construct();
8	\$this->load->model('M_training_data');
9	}
10	//fungsi yang melibatkan ajax untuk menampilkan data latih yang ada pada database melalui pemanggilan dari kelas M_training_data supaya
11	//ditampilkan pada halaman utama dengan bentuk tabel
12	function ajaxShow()
13	{
14	\$list = \$this->M_training_data->tampil();
15	\$data = array();
16	\$no = \$_POST['start']; //variabel \$no digunakan untuk penomoran pada isi

	dari pada variabel \$List
17	foreach (\$list as \$key) {
18	\$no++; //isi dari \$no selalu bertambah 1 setiap kali perulangan sebanyak jumlah dari data latih
19	\$row = array();
20	\$row[] = \$key->kalimat;
21	\$row[] = \$key->status;
22	\$row[] = 'id . ') "><i class="fa fa-edit"></i>
23	<button class="btn btn-danger btn-xs" title="Hapus" onclick="hapus(' . \$key->id . ') "><i class="fa fa-trash"></i></button>
24	';
25	\$data[] = \$row; //hasil dari isi array row[] disimpan pada array \$data secara kolektif dimana isinya terdiri dari kalimat dan status
26	
27	}
28	\$output = array("draw" => \$_POST['draw'],
29	"recordsTotal" => \$this->M_training_data->count_all(),//recordTotal melakukan pemanggilan fungsi count_all pada kelas M_training_data untuk menghitung keseluruhan jumlah baris data latih yang akan ditampilkan pada kelas view
30	"recordsFiltered" => \$this->M_training_data->count_filtered(),//recordTotal melakukan pemanggilan fungsi count_filtered pada kelas M_training_data untuk menghitung jumlah baris data latih yang ditampilkan pada setiap sheet yang ada pada kelas view, 1 sheet berisikan 10 dokumen
31	"data" => \$data); //\$output menyimpan secara kolektif dari nilai nilai yang sudah dideklarasikan, kemudian di teruskan melalui json untuk diproses di kelas view.
32	
33	echo json_encode(\$output);
34	}
35	
36	}

Kelas view akan ditunjukkan sebagian potongan kode yang berfungsi untuk memanggil kelas C_Training_data.php dan menampilkan hasil pemanggilan berupa tabel dokumen twitter.

Kode Program 5.3 modelling_data.php – get data tabel

1	
2	//kode program ini untuk menyusun model data ditampilkan pada program
3	table = \$('#data_barang').DataTable({
4	"scrollY": false,
5	"scrollX": true,
6	"processing": true, //Feature control the processing indicator.
7	"serverSide": true, //Feature control DataTables' server-side processing mode.
8	"order": [], //Initial no order.
9	
10	// Load data for the table's content from an Ajax source
11	"ajax": {
12	"url": "<?php echo site_url('C_Training_data/ajaxShow');?>",
13	"type": "POST"
14	},
15	
16	//Set column definition initialisation properties.
17	"columnDefs": [
18	{
19	"targets": [-1], //last column
20	"orderable": false, //set not orderable
21	},
22],
23	
24	});
25	

5.1.2 Implementasi Preprocessing

Proses ini di panggil dengan menggunakan kelas controller Text_processing.php.

Kode Program 5.4 Text_processing.php

1	<?php
2	defined('BASEPATH') OR exit('No direct script access allowed');

3	
4	class Text_processing extends CI_Controller {
5	
6	function __construct(){
7	parent::__construct();
8	\$this->load->model('M_processing');
9	}
10	
11	public function index() {
12	\$this->load->view('text_processing');
13	}
14	//fungsi untuk tokenisasi pada data untuk difilter dengan stoplist, normalisasi, dan stemming
15	function get_data_stem_stop()
16	{
17	\$this->load->library('Stemming_sastrawi'); //meload library sastrawi untuk proses stemming
18	\$stemmerFactory = new \Sastrawi\Stemmer\StemmerFactory();
19	\$stemmer = \$stemmerFactory->createStemmer(); //inisiasi variabel \$stemmer untuk memanggil library sastrawi
20	
21	\$data_training = \$this->M_processing->get_data_training(); //inisiasi variabel \$data_training dari tabel data training melalui kelas model M_processing
22	\$data = array();
23	foreach (\$data_training as \$key) {
24	\$words = \$this->M_processing->clear_tag(\$key->kalimat);
25	\$stemmed = \$stemmer->stem(\$words);
26	\$stopped_kalimat = \$this->M_processing->stop_removal(\$stemmed);
27	//menyimpan nilai data yang sudah diproses kedalam array %data[]
28	\$data[] = \$key->id."[]". \$stopped_kalimat."[]". \$key->status;
29	}
30	\$this->db->empty_table("likelihood");
31	echo json_encode(\$data); //mengembalikan data untuk dihantarkan ke kelas view text_processing
32	}

33	//fungsi untuk mendapatkan kata unik pada data yang sudah di proses pada fungsi get_data_stem_stop
34	function get_kata_unik()
35	{
36	\$unik = \$this->input->post('words'); //menangkap atau mengambil data dari proses sebelumnya yang di inisiasikan sebagai words yang berisikan kata-kata unik hasil prprocessing
37	
38	\$unik = explode(" ", \$unik); //memecah kata unik dengan karakter putih / spasi
39	\$unik = array_filter(\$unik); //membersihkan \$unik dari kata unik yang terduplikasi selama proses berlangsung
40	\$words = array();
41	foreach (\$unik as \$key) {
42	if (!in_array(\$key, \$words)){ //penkondisian untuk pengecekan nilai variabel \$key terhadap \$word, dan jika benar maka
43	\$words[] = \$key; //nilai daripada \$key akan dimasukkan kedalam array \$words[]
44	}
45	}
46	\$data['words'] = \$words; //menyimpan nilai word yang berisikan kata unik kedalam array \$data
47	\$data['docs'] = \$this->input->post('docs');
48	echo json_encode(\$data); //mengembalikan data untuk dihantarkan ke kelas view text_processing
49	}
50	function term_frequency() //fungsi untuk pencarian tf-biner
51	{
52	\$data = \$this->input->post('data');
53	
54	\$data['tf'] = array(); //inisiasi array \$data yang nantinya akan di isi dengan nilai tf tiap kata
55	
56	foreach (\$data['words'] as \$key) {
57	foreach (\$data['docs'] as \$key_satu) {
58	\$indexing = explode("[]", \$key_satu);
59	\$kemunculan = 0;

60	\$kalimat_ = explode(" ", \$indexing[1]);
61	
62	foreach (\$kalimat_ as \$kalimat) {
63	if (\$kalimat==\$key) { //jika pada variable \$kalimat ditemukan isi dari \$key yang merupakan kata unik, maka
64	\$kemunculan = 1; //variabel kemunculan akan diberi nilai 1 untuk kata unik yang sedang diproses saat itu
65	}
66	}
67	
68	\$row[] = \$indexing[0]." ".\$indexing[2]." ".\$key." ".\$kemunculan;
69	}
70	\$data['tf'] = \$row; //memasukan nilai \$row yang berisikan informasi dari kata unik, status, dan jumlah kemunculannya pada array \$data['tf']
71	}
72	
73	echo json_encode(\$data); //mengembalikan data untuk dihantarkan ke kelas view text_processing
74	}

Kelas Model digunakan untuk mengolah data yang dipanggil oleh controller untuk disimpan di database maupun mengambil data dari database.

Kode Program 5.5 M_processing.php

1	<?php if (! defined('BASEPATH')) exit('No direct script access allowed');
2	
3	class M_processing extends CI_Model {
5	var \$table1 = 'data_training';
7	var \$column = array("id","kalimat","status");
8	
9	//fungsi yang dipanggil saat controller mulai memproses preprocessing, fungsi ini untuk menghapus karakter yang tidak penting
10	function stop_removal(\$data)
11	{
13	\$words = \$this->removal(); //pemanggilan fungsi removal() untuk proses penghilangan kata yang tidak penting sesuai library pada fungsi removal()

14	\$words = preg_replace('/\b('implode(' ',\$words).')\b/', '', \$data); //setelah itu dicocokkan dengan isi dari data latih (\$data) untuk penghilangan kata
15	\$words = \$this->kata_alay(\$words); //memanggil fungsi_alay untuk memproses normalisasi bahasa alay menjadi kata baku (prosesnya ada di fungsi selanjutnya).
16	
17	return \$words;
18	}
19	
20	//fungsi untuk normalisasi bahasa alay menjadi bahasa yang baku sesuai kbbi
21	function kata_alay(\$words)
22	{
23	\$this->db->from('kata_alay');
24	\$kata_alay = \$this->db->get()->result();//mengambil isi dari tabel kata alay, untuk diproses dalam perulangan supaya mengganti keseluruhan kata yang alay nantinya
25	
26	foreach (\$kata_alay as \$key) {
27	\$words = preg_replace('/\b('.\$key->alay.')\b/', \$key->kbbi, \$words); //proses menggantikan kata alay yang ditemukan pada setiap data latih untuk di rubah katanya menjadi kata yang baku sesuai kbbi
28	}
29	
30	return \$words; //mengembalikan nilai variabel \$word untuk diproses di kelas controller
31	}
32	///fungsi untuk menghilangkan hastag, link, dan username pada dokumen
33	function clear_tag(\$words)
34	{
35	\$regex_at = '/@S+\s*/'; //inisiasi regex
36	\$regex_hash = '/#S+\s*/';
37	
38	\$words = preg_replace(\$regex_at, '', \$words); //mreplace kata-kata pada \$words dengan proses regex
39	\$words = preg_replace(\$regex_hash, '', \$words);
40	
41	\$words = preg_replace('/\b(https? ftp file):\\V[-A-Z0-9+&@#\\V%?=_~ \$!.,;]*[A-Z0-9+&@#\\V%=_~ \$/i', '', \$words);

42	\$words = preg_replace('/\b(http? ftp file):\\V[-A-Z0-9+&@#\\/%?=_ \$!.,;]*[A-Z0-9+&@#\\/%?=_ \$/i', '', \$words);
43	
44	return \$words;
45	}
46	
47	function get_data_training()
48	{
49	\$this->db->from(\$this->table1); //memanggil isi dari tabel data training
50	return \$this->db->get()->result();
51	}
52	
53	//fungsi ini akan dipanggil oleh fungsi stop_removal
54	function removal()
55	{ //array \$word berisikan kata-kata yang pasti dihapuskan saat ditemukan pada isi dari data latih
56	\$words = array(
57	'yang', 'untuk', 'pada', 'ke', 'para', 'antara', 'dua',
58	'jika', 'dan', 'ini', 'karena', 'karna',
59	'kepada', 'oleh', 'saat', 'sekitar',
60	'bagi', 'serta', 'di', 'dari', 'sebagai', 'masih', 'hal', 'ketika', 'adalah',
61	'itu', 'dalam', 'bahwa', 'atau', 'dengan', 'akan', 'juga',
62	'terhadap', 'secara', 'lain',
63	'begitu', 'mengapa', 'kayak', 'kenapa', 'yaitu', 'yakni', 'itulah', 'maka',
64	'tentang', 'dimana', 'kemana', 'pula', 'sambil',
65	'kah', 'pun', 'sampai', 'selagi', 'apakah', 'Shiiittttt',
66	'seolah', 'seraya', 'seterusnya', 'agak', 'wkwkwkwkwkwkw',
67	'dsb', 'dst', 'dll', 'anu', 'demikian', 'beh',
68	'juga', 'nggak', 'mari', 'nanti', 'melainkan', 'oh', 'ok', 'seharusnya', 'harusnya', 'sebetulnya',
69	'setiap', 'setidaknya', 'sesuatu', 'saja', 'toh', 'ya', 'walau', 'tolong', 'eh', 'dech', 'anjay',
70	'tentu', 'amat', 'apalagi', 'bagaimanapun', 'kata', '#', 'www', '@', '#', 'mustika', 'ratu', 'com', 'mustikaratu',
71	'kalo', 'kalau', '=', '!', '-', 'T', '_', '(', ')', ':', 'euy', 'hahhahaaaa', ' ahahhaaaa',

	'wkwkwkwkwk', 'alah',
72	'viva', 'baby', 'oil', 'johnsons', 'jadi', 'tiap', 'endooolllllll', '&', 'sama', 'wkwkwkwkwkw', 'Alaaamakkkkk',
73	'wkwk', 'cuma', 'deh', 'ini', 'siapa', 'bukan', 'lah', 'emang', 'no', 'sebut', 'bahhhhh', 'wkwkwkwk',
74	'bahkan', 'buat', 'di', 'ke', 'biar', 'Mustikaratu', 'mustika', 'ratu', 'gimana', 'ka', 'kakak', 'heh',
75	'kalau', 'apa', 'yah', 'malah', 'sih', 'kok', 'tuh', 'beb', 'loh', 'pokoknya',
76	'uuh', 'terus', 'ama', 'lah', 'hehe', 'aduh', 'tanpa', 'agh', 'cream', 'cc', 'lipstik', 'dah', 'ih', 'kali',
77	'ya', 'RT', 'wardah', 'yeahh', 'hahhh', 'huhh', 'lha', 'mah', 'hemmm', 'wuihh', 'lah', 'ihh', 'heuheu',
78	'ckckckckckcccc',
79);
80	return \$words;
81	}
82	}

Sedangkan untuk kelas view nya digunakn untuk melewati data hasil processing untuk dibawah ke proses selanjutnya.:

Kode Program 5.6 processing.php

1	
2	\$('#btn-generate-mulai').click(function(){
3	\$('#btn-generate-mulai').html('<i class="fa fa-spinner fa-spin"></i>');
4	\$.ajax({
5	url : "<?php echo site_url('Text_processing/get_data_stem_stop')?>",
6	type: "GET",
7	dataType: "JSON",
8	progress: function(e) {
9	// menghitung panjang
10	if(e.lengthComputable) {
11	//kalkulasi presentasi yang terload
12	var pct = (e.loaded / e.total) * 100;
13	
14	//kunci percentage terload

15	\$.proses-data").html("Sedang mengambil data..." + pct);
16	}
17	//jika Content-Length tidak terset
18	else {
19	console.warn('Content Length not reported!');
20	}
21	},
22	success: function(data)
23	{
24	
25	if (data) {
26	\$.pengambilan-data").html(data.length + " Data berhasil diambil menjadi kata dasar dan kata tidak penting telah terhapus <i class='fa fa-check'></i>");
27	// stop_removal(data);
28	
29	};
30	
31	var newHTML = [];
32	var semua_Kalimat = [];
33	for (var i = data.length - 1; i >= 0; i--) {
34	data[i]
35	var spliter = data[i].split(" []")
36	
37	semua_Kalimat.push(spliter[1]);
38	//
	newHTML.push("<tr><td>" + spliter[0] + "</td><td>" + spliter[1] + "</td><td>" + spliter[2] + "</td></tr>");
39	};
40	
41	// \$.tabel-data-training").html(newHTML.join(""));
42	get_kata_unik(data, semua_Kalimat.join(" "));
43	
44	
45	},
46	error: function (jqXHR, textStatus, errorThrown)

47	{
48	alert('Error get data from ajax');
49	}
50	});
51	}); //fungsi untuk mendapat kata unik setelah menerima //instruksi dari kelas controller, data yang diproses dari variabel docs dan variabel words
52	function get_kata_unik(data, semua_Kalimat)
53	{
54	\$.ajax({
55	url : "<?php echo site_url('Text_processing/get_kata_unik')?>",
56	type: "POST",
57	data: {'docs': data, 'words': semua_Kalimat},
58	dataType: "JSON",
59	progress: function(e) {
60	// menghitung panjang data
61	if(e.lengthComputable) {
62	//kalkulasi presentasi yang terload
63	var pct = (e.loaded / e.total) * 100;
64	
65	//kunci percentage terload
66	\$("#pengambilan-kata-unik").html("Sedang mengambil data..." + pct);
67	}
68	//jika Content-Length tidak ter set
69	else {
70	console.warn('Content Length not reported!');
71	}},
72	success: function(data)
73	{
74	\$("#pengambilan-kata-unik").html("Kata unik berhasil diambil <i class='fa fa-check'></i>");
75	\$('.btn-generate-mulai').html('<i class="fa fa-spinner fa-spin"></i> mengihtung nilai tf dan df..');
76	term_frequency(data);
77	},
78	error: function (jqXHR, textStatus, errorThrown) {

5.1.3 Implementasi TF-Biner

Bagian kelas controller `Text_processing.php` akan mengambil data pada variabel `$data` hasil olahan proses sebelumnya, potongan kode berupa fungsi `term_frequency` bertugas untuk mendapatkan nilai tf-biner pada masing-masing kata unik yang ada pada dokumen yang terkumpulkan.

Kode Program 5.7 `Text_processing.php` - function `term_frequency()`

1	
2	<code>function term_frequency() //fungsi untuk pencarian tf-biner</code>
3	<code>{</code>
4	<code>\$data = \$this->input->post('data'); //inisiasi array \$data yang nantinya akan di</code> <code>isi dengan nilai tf tiap kata</code>
5	<code>\$data['tf'] = array();</code>
6	
7	<code>foreach (\$data['words'] as \$key) {</code>
8	
9	<code>foreach (\$data['docs'] as \$key_satu) {</code>
10	<code>\$indexing = explode("[]", \$key_satu);</code>
11	<code>\$kemunculan = 0;</code>
12	<code>\$kalimat_ = explode(" ", \$indexing[1]);</code>
13	
14	<code>foreach (\$kalimat_ as \$kalimat) {</code>
15	<code>if (\$kalimat==\$key) { //jika pada variable \$kalimat ditemukan isi dari</code> <code>\$key yang merupakan kata unik, maka</code>
16	<code>\$kemunculan = 1; //variabel kemunculan akan diberi nilai 1 untuk</code> <code>kata unik yang sedang diproses saat itu</code>
17	<code>}</code>
18	<code>}</code>
19	<code>\$row[] = \$indexing[0]." ".\$indexing[2]." ".\$key." ".\$kemunculan;</code>
20	<code>}</code>
21	<code>\$data['tf'] = \$row; //memasukan nilai \$row yang berisikan informasi dari</code> <code>kata unik, status, dan jumlah kemunculannya pada array \$data['tf']</code>
22	<code>}</code>
23	<code>echo json_encode(\$data); } //mengembalikan data untuk dihantarkan ke kelas</code> <code>view text_processing</code>

Sedangkan untuk kelas view processing.php yang bertugas untuk memanggil fungsi diatas, dan mendapatkan data dari proses sebelumnya, serta meneruskan ke proses selanjutnya pada bagian berikut.

Kode Program 5.8 processing.php - function term_frequency(data)

1	
2	
3	function term_frequency(data)
4	{
5	\$.ajax({
6	url : "<?php echo site_url('Text_processing/term_frequency')?>",
7	type: "POST",
8	data: {'data': data},
9	dataType: "JSON",
10	progress: function(e) {
11	//Menghitung Panjang Data
12	if(e.lengthComputable) {
13	//kalkulasi persentase terload
14	var pct = (e.loaded / e.total) * 100;
15	
16	//kunci persentase terload
17	
18	\$(".pengambilan-kata-unik").html("Sedang mengambil data..." + pct);
19	\$(".pengambilan-term-frequency").html("Perhitungan frekuensi kata sukses <i class='fa fa-check'></i>");
20	
21	}
22	//this usually happens when Content-Length isn't set
23	else {
24	console.warn('Content Length not reported!');
25	}
26	},
27	success: function(data)
28	{
29	\$(".pengambilan-kata-unik").html("Kata unik berhasil diambil <i class='fa

	fa-check'></i>");
30	\$('.btn-generate-mulai').html('<i class="fa fa-spinner fa-spin"></i> menghitung nilai entropy');
31	get_entropy(data);
32	},
33	error: function (jqXHR, textStatus, errorThrown)
34	{
35	} }); }

5.1.4 Implementasi Information Gain

Pada bagian proses *Information Gain* ini dibagi menjadi 3 bagian yang mendasar, pertama terdapat proses untuk mengambil nilai entropy total, kedua mendapatkan nilai partisi ada dan partisi tidak ada pada kelas positif dan kelas negatif, dan ketiga mencari nilai *Information Gain* pada setiap term atau kata unik.

Proses mendapatkan nilai entropy total adalah dengan memanggil fungsi `get_entropy()` yang ada pada `Text_processing.php` sebagai berikut :

Kode Program 5.9 Text_processing.php - get_entropy()

1	//fungsi untuk mencari nilai entropy total pada dokumen
2	function get_entropy() {
3	\$data = \$this->input->post('data');
4	\$positive = 0;
5	\$negative = 0;
6	foreach (\$data as \$key) {
7	\$dokumen = explode("[]", \$key);
8	if (\$dokumen[2]=="positif") { //jika pada \$dokumen indeks 2 nilainya sama dengan nilai "positif", maka
9	\$positive = \$positive+1; //variable \$positif ditambahkan +1 pada nilai sebelumnya
10	}
11	Else
12	{ // jika tidak, maka yang ditambahkan +1 adalah variabel \$negatif
13	\$negative = \$negative+1;
14	}
15	}

16	
17	\$total = \$positive+\$negative;
18	\$entropy_positive = \$positive/\$total*log(\$positive/\$total, 2);
19	\$entropy_negative = \$negative/\$total*log(\$negative/\$total, 2);
20	
21	\$entropy_total = -(\$entropy_positive + \$entropy_negative);
22	echo json_encode(\$entropy_total);
23	}

Sedangkan untuk kelas view processingnya sebagai berikut, melewati data untuk diproses selanjutnya, yaitu proses mencari nilai entropyA(D), bagian kelas view pula yang akan melewati dan meneruskan ke proses-proses selanjutnya :

Kode Program 5.10 processing.php – get_entropy(data)

1	//fungsi untuk mendapat nilai entropy total setelah menerima
2	//instruksi dari kelas controller, data yang diproses dari array data['docs']
3	function get_entropy(data)
4	{
5	\$.ajax({
6	url : "<?php echo site_url('Text_processing/get_entropy')?>",
7	type: "POST",
8	data: {'data': data['docs']},
9	dataType: "JSON",
10	progress: function(e) {
11	//menghitung panjang data
12	if(e.lengthComputable) {
13	//kalkulasi persentase terload
14	var pct = (e.loaded / e.total) * 100;
15	
16	//kunci persentase terload
17	
18	\$(".pengambilan-kata-unik").html("Sedang mengambil data..." + pct);
19	}
20	//this usually happens when Content-Length isn't set
21	else {

22	console.warn('Content Length not reported!');
23	}
24	},
25	success: function(entropy_total)
26	{
27	\$("#pengambilan-entropy").html("Perhitungan entropy total sukses <i class='fa fa-check'></i>");
28	\$('#.btn-generate-mulai').html('<i class="fa fa-spinner fa-spin"></i> menghitung nilai entropy');
29	get_entropy_partisi_P_N(entropy_total, data);
30	},
31	error: function (jqXHR, textStatus, errorThrown)
32	{
33	alert("error brooo");
34	} }); }

Proses selanjutnya yang kedua, dengan memanggil fungsi `get_entropy_partisi_P_N()` supaya pada setiap dokumen yang ada pada kelasnya didapatkan nilai entropy partisi ada dan partisi tidak ada.

Kode Program 5.11 Text_processing.php - `get_entropy_partisi_P_N()`

1	
2	//fungsi untuk mencari nilai entropy partisi ada dan tidak ada berdasarkan hasil perhitungan tf-biner
3	function get_entropy_partisi_P_N()
4	{
5	\$kata_unik = \$this->input->post('kata_unik'); //inisiasi \$kata_unik dengan keywordkata_unik untuk pemanggilan data yang sudah di definisikan pada kelas view
6	
7	\$tf = \$this->input->post('tf');
8	\$dokumen = \$this->input->post('dokumen');
9	\$hitung_dokumen = count(\$dokumen); //menghitung jumlah dokumen yang di jadikan data latih dan dimasukan nilainya kedalam \$hitung_dokumen
10	\$data_entropy = array();
11	
12	foreach (\$kata_unik as \$kata_unik_) {

13	\$ada_positive = 0;
14	\$ada_negative = 0;
15	
16	\$tdk_positive = 0;
17	\$tdk_negative = 0; //memberikan nilai awal pada variabel \$tdk_negative berupa angka 0 yang berarti tidak ada nilai nya dan bertipe integer.
18	
19	foreach (\$tf as \$tf_) {
20	\$tf_explode = explode(" ", \$tf_);
21	if (\$tf_explode[1]=="positif" && \$tf_explode[2] == \$kata_unik_ && \$tf_explode[3]=="1") { //proses pengkondisian jika dalam suatu dokumen berkelas positif memiliki kata unik yang sama dengan isi dari variabel \$kata_unik_, dan ditemukan nilai kemunculannya, maka
22	\$ada_positive = \$ada_positive + 1; //variabel \$ada_positif yang sebelumnya ditambahkan 1 untuk memperbarui nilai \$ada_positive
23	}
24	elseif (\$tf_explode[1]=="negatif" && \$tf_explode[2] == \$kata_unik_ && \$tf_explode[3]=="1") {
25	\$ada_negative = \$ada_negative + 1;
26	}
27	elseif (\$tf_explode[1]=="positif" && \$tf_explode[2] == \$kata_unik_ && \$tf_explode[3]=="0") { //proses pengkondisian jika dalam suatu dokumen berkelas positif memiliki kata unik yang sama dengan isi dari variabel \$kata_unik_, dan tidak ditemukan nilai kemunculannya, maka
28	\$tdk_positive = \$tdk_positive + 1; //variabel \$tdk_positive yang sebelumnya ditambahkan 1 untuk memperbarui nilai \$tdk_positive
29	}
30	elseif (\$tf_explode[1]=="negatif" && \$tf_explode[2] == \$kata_unik_ && \$tf_explode[3]=="0") {
31	\$tdk_negative = \$tdk_negative + 1;
32	}
33	}
34	
35	\$total_ada = \$ada_positive+\$ada_negative;
36	\$total_tdk = \$tdk_positive+\$tdk_negative;
37	
38	if (\$total_ada!=0) { //pengkondisian jika jumlah total kata tertentu ada pada

	keseluruhan dokumen tidak sama dengan 0 / ada nilainya, maka
39	\$logging_ada_positive = \$ada_positive/\$total_ada; //menghitung nilai \$logging_ada_positif dimana pembagian jumlah kata tertentu yang ada di dokumen positif dengan jumlah total kata tertentu pada keseluruhan dokumen
40	\$logging_ada_negative = \$ada_negative/\$total_ada;
41	}
42	
43	else{
44	\$logging_ada_positive = 0; //jika tdak ada isinya, maka nilai \$logging_ada_positif = 0
45	\$logging_ada_negative = 0;
46	}
47	
48	if (\$logging_ada_positive!=0) { //setelah itu jika \$logging_ada_positive tidak bernilai 0, maka
49	\$entropy_ada_positif = -(\$ada_positive/\$total_ada*log(\$logging_ada_positive,2));
50	}
51	else {
52	\$entropy_ada_positif = 0; //jika tidak, maka \$entropy_ada_positif langsung dinilai 0, tanpa ada perhitungan sesuai rumus
53	}
57	
54	if (\$logging_ada_negative!=0) {
55	\$entropy_ada_negatif = - (\$ada_negative/\$total_ada*log(\$logging_ada_negative,2));
56	}
57	else {
58	\$entropy_ada_negatif = 0;
59	}
60	
61	\$total_entropy_ada = \$entropy_ada_positif + \$entropy_ada_negatif; //setelah didapati entropi ada dihitung dengan menambahkan kedua variabel disamoing
62	
63	if (\$total_tdk!=0) {
64	\$logging_tdk_positive = \$tdk_positive/\$total_tdk;

65	\$logging_tdk_negative = \$tdk_negative/\$total_tdk;
66	}
67	else {
68	\$logging_tdk_positive = 0;
69	\$logging_tdk_negative = 0;
70	}
71	
72	if (\$logging_tdk_positive!=0) {
73	\$entropy_tdk_positif = -(\$tdk_positive/\$total_tdk*log(\$logging_tdk_positive,2));
74	}
75	Else
76	{
77	\$entropy_tdk_positif = 0;
78	}
79	
80	if (\$logging_tdk_negative!=0) {
81	\$entropy_tdk_negatif = -(\$tdk_negative/\$total_tdk*log(\$logging_tdk_negative,2));
82	}
83	
84	else {
85	\$entropy_tdk_negatif = 0;
86	}
87	
88	\$total_entropy_tdk = \$entropy_tdk_positif + \$entropy_tdk_negatif;
89	
90	\$entropy_ad = (((\$total_ada/\$hitung_dokumen)*\$total_entropy_ada)+((\$total_tdk/\$hitung_dokumen*\$total_entropy_tdk));
91	
92	\$row[\$total_entropy_ada." ".\$total_entropy_tdk." ".\$entropy_ad." ".\$kata_unik_;
93	
94	\$data_entropy = \$row;

95	
96	
97	echo json_encode(\$data_entropy); //mengembalikan data untuk dihantarkan ke kelas view text_processing
98	}

Untuk potongan kode Kelas Viewnya adalah sebagai berikut :

Kode Program 5.12 processing.php - get_entropy_partisi_P_N(entropy_total, data)

1	//fungsi untuk mendapat nilai entroy partisi ada, dan tidak ada sekliigus mencari nilai entropyA(D) setelah menerima
2	//instruksi dari kelas controller, data yang diproses dari array data['docs'], variabel entropy_total, array data['words'], //dan array data['tf']
3	function get_entropy_partisi_P_N(entropy_total, data)
4	{
5	\$.ajax({
6	url : "<?php echo site_url('Text_processing/get_entropy_partisi_P_N')?>",
7	type: "POST",
8	data: {'dokumen': data['docs'], 'entropy_total': entropy_total, 'kata_unik': data['words'], 'tf': data['tf']},
9	dataType: "JSON",
10	progress: function(e) {
11	//make sure we can compute the length
12	if(e.lengthComputable) {
13	//calculate the percentage loaded
14	var pct = (e.loaded / e.total) * 100;
15	
16	//log percentage loaded
17	
18	\$(".pengambilan-kata-unik").html("Sedang mengambil data..." + pct);
20	}
21	//this usually happens when Content-Length isn't set
22	else {
23	console.warn('Content Length not reported!');

24	}
25	},
26	success: function(data_entropy)
27	{
28	\$("#pengambilan-entropy-partisi").html("Perhitungan entropy partisi sukses <i class='fa fa-check'></i>");
29	\$('#btn-generate-mulai').html('<i class="fa fa-spinner fa-spin"></i> menghitung info gain');
30	info_gain(entropy_total, data, data_entropy);
31	},
32	error: function (jqXHR, textStatus, errorThrown)
33	{
34	
35	alert("error brooo");
36	}}); }

Proses yang terakhir adalah mencari nilai info gainnya, didapatkan dengan mengurangi hasil dari proses fungsi get_entropy yang merupakan nilai entropy total dengan hasil tiap entropy partisi ada dan partisi tidak ada.

Kode Program 5.13 Text_processing.php – info_gain()

1	
2	//fungsi untuk mencari nilai info gain dari tiap fitur/kata, nilainya didapat dengan mengurangi nilai entorpy total dengan entropyA(D)
3	function info_gain()
4	{
5	\$data = \$this->input->post('data');
6	\$entropy_total = \$this->input->post('entropy_total');
7	\$data_entropy = \$this->input->post('data_entropy');
8	
9	\$info_gain = array();
10	foreach (\$data_entropy as \$data_entropy_) {
11	\$entropy = explode(" ", \$data_entropy_); //memisahkan data pada \$data_entropy hasil proses fungsi sebelumnya, dengan tanda ' ' dan kemudian di simpan pada \$entropy
12	

13	\$info_gain_ = \$entropy_total-\$entropy[2]; //setelah dilakukan pemisahaan data pada arraay \$data_entropy, nilai \$info_gain dapat dicari dengan mengurangi nilai \$entropy_total dengan \$entropy pada indeks array ke 2
14	\$info_gain[] = \$info_gain_" ".\$entropy[3]; //nilai dar \$info_gain_ dimasukkan pada array \$info_gain[]
15	}
16	echo json_encode(\$info_gain);
17	}

Potongan kode untuk kelas view fungsi info_gain sebagai berikut :

Kode Program 5.14 processing – info_gain(entropy_total, data, data_entropy)

1	//fungsi untuk mendapat nilai info gain setelah menerima
2	//instruksi dari kelas controller, data yang diproses dari array data['docs'], variabel entropy_total, dan data_entropy
3	function info_gain(entropy_total, data, data_entropy)
4	{
5	\$.ajax({
6	url : "<?php echo site_url('Text_processing/info_gain')?>",
7	type: "POST",
8	data: {'data': data, 'entropy_total': entropy_total, 'data_entropy': data_entropy },
9	dataType: "JSON",
10	progress: function(e) {
11	//menghitung panjang data
12	if(e.lengthComputable) {
13	//kalkulasi percentase terload
14	var pct = (e.loaded / e.total) * 100;
15	
16	//kunci percentase terload
17	\$(".pengambilan-kata-unik").html("Sedang mengambil data..." +pct);
18	}
19	//this usually happens when Content-Length isn't set
20	else {
21	console.warn('Content Length not reported!');
22	}

23	},
24	success: function(info_gain)
25	{
26	\$(".pengambilan-entropy-partisi").html("Perhitungan entropy partisi sukses <i class='fa fa-check'></i>");
27	\$('.btn-generate-mulai').html('<i class="fa fa-spinner fa-spin"></i> menghapus kata unik');
28	frekuensi_dokumen2(info_gain, data);
29	},
30	error: function (jqXHR, textStatus, errorThrown)
31	{
32	alert("error brooo");
33	}
34	}); }

5.1.5 Implementasi *Naive Bayes Classifier*

Proses *Naive Bayes Classifier* digunakan untuk mendapatkan nilai probabilitas dari tiap dokumen untuk diklasifikasikan sebagai kelas positif maupun negatif. Proses *Naive Bayes Classifier* melibatkan 4 proses, pertama mendapatkan nilai term frekuensi pada tiap term yang ada pada dokumen dokumen yang berkelas tertentu untuk digunakan mencari nilai likelihood nantinya, kedua mencari likelihood dari tiap term dan menyimpannya dalam database, ketiga mencari nilai prior dan menyimpannya dalam database, dan yang keempat probabilitas dengan mengkalikan likelihood dengan priornya.

Proses yang pertama, mendapatkan nilai banyaknya dokumen term frekuensi pada tiap term yang ada pada dokumen dokumen yang berkelas tertentu. Pada penelitian ini terdapat kelas positif dan negatif, oleh sebab itu nilai term frekuensi akan di total dan dipisahkan menjadi 2 bagian, total term frekuensi pada dokumen berkelas positif dan total term frekuensi pada dokumen berkelas negatif.

Kode Program 5.15 `Text_processing.php` – `frekuensi_term2()`

1	//fungsi untuk digunakan mencari nilai term setiap fitur yang nantinya dipakai untuk mencari nilai likelihoodnya
2	function frekuensi_term2()
3	{
4	\$data = \$this->input->post('data');
5	\$kata_unik_baru = \$this->input->post('info_gain');

6	
7	\$frekuensi_baru = array();
8	foreach (\$kata_unik_baru as \$kata_unik_) {
9	\$data_gain = explode(" ", \$kata_unik_);
10	
11	foreach (\$data as \$docs) {
12	
13	\$indexing = explode("[]", \$docs);
14	\$kalimat = explode(" ", \$indexing[1]);
15	\$jumlah_kata = 0;
16	
17	foreach (\$kalimat as \$kata) {
18	if (\$kata==\$data_gain[1]) {
19	\$jumlah_kata = 1;
20	}
21	}
22	\$frekuensi_baru[\$data_gain[1]." ".\$jumlah_kata." ".\$indexing[2]." ".\$data_gain[0];
23	}
24	}
25	echo json_encode(\$frekuensi_baru);
26	}
27	

Kelas view processing.php akan memanggil fungsi pada kelas diatas dengan menggunakan potongan kode berikut.

Kode Program 5.16 processing.php – frekuensi_term2(info_gain, data)

1	
2	
3	function frekuensi_term2(info_gain, data)
4	{
5	th = document.getElementById("nilai-threshold").value;
6	\$.ajax({
7	url : "<?php echo site_url('Text_processing/frekuensi_term2')?>",

8	type: "POST",	
9	data: {'data': data['docs'], 'info_gain': info_gain},	
10	dataType: "JSON",	
11	progress: function(e) {	
12	//menghitung panjang data	
13	if(e.lengthComputable) {	
14	//kalkulasi percentase terload	
15	var pct = (e.loaded / e.total) * 100;	
16		
17	//log percentage loaded	
18	\$(".pengambilan-kata-unik").html("Sedang mengambil data..." + pct);	
19		
20	}	
21	//jika Content-Length tidak terdeteksi	
22	else {	
23	console.warn('Content Length not reported!');	
24	}	
25	},	
26	success: function(term_frequency2)	
27	{	
28	\$(".pengambilan-entropy-partisi").html("Perhitungan entropy partisi sukses <i class='fa fa-check'></i>");	
29	\$(".btn-generate-mulai").html("<i class='fa fa-spinner fa-spin'></i> selesai");	
30	likelihood(info_gain, data, term_frequency2);	
31	},	
32	error: function (jqXHR, textStatus, errorThrown)	
33	{	
34	alert("error brooo");	
35	}}); }	

Proses yang kedua, menggunakan fungsi likelihood() yang ada pada kelas Naive_bayes.php sebagai controller lainnya, namun lebih dispesifikan untuk mengklasifikasikan dokumen, dan untuk menyimpan nilai likelihood dapat menggunakan fungsi simpan_likelihood() pada kelas controller Naive_bayes.php melalui kelas model M_naivebayes.php untuk diteruskan ke database.

Kode Program 5.17 Naive_bayes.php - function likelihood()

1	<?php
2	defined('BASEPATH') OR exit('No direct script access allowed');
3	
4	class Naive_bayes extends CI_Controller {
5	
6	function __construct(){
7	parent::__construct();
8	\$this->load->model('M_processing');
9	}
10	//fungsi untuk mencari nilai likelihood
11	function likelihood()
12	{
13	\$kata_unik_lama = \$this->input->post('data');
14	\$kata_unik_baru = \$this->input->post('kata_unik_baru');
15	\$term_frequency2 = \$this->input->post('term_frequency2');
16	\$threshold = \$this->input->post('threshold');
17	
18	\$jumlah_kata_unik_baru = count(\$kata_unik_baru);
19	\$likelihood = array();
20	
21	\$total_semua_positive = 0;
22	\$total_semua_negative = 0;
23	
24	foreach (\$kata_unik_lama as \$kata_unik) {
25	\$total_positive = 0;
26	\$total_negative = 0;
27	
28	\$all_negative = 0;

29	\$all_positive = 0;
30	foreach (\$term_frequency2 as \$term) {
31	\$term = explode(" ", \$term);
32	
33	if (\$term[0]==\$kata_unik && \$term[1]=="1" && \$term[2]=="positif") {
34	\$total_positive=\$total_positive+1;
35	}
36	elseif (\$term[0]==\$kata_unik && \$term[1]=="1" && \$term[2]=="negatif") {
37	\$total_negative=\$total_negative+1;
38	}
39	}
40	
41	\$total_semua_positive = \$total_semua_positive + \$total_positive;
42	\$total_semua_negative = \$total_semua_negative + \$total_negative;
43	}
44	
45	foreach (\$kata_unik_baru as \$kata_unik_) {
46	\$kata_unik_ = explode(" ", \$kata_unik_);
47	\$kata_unik = \$kata_unik_[1];
48	\$info_gain = \$kata_unik_[0];
49	
50	\$total_positive = 0;
51	\$total_negative = 0;
52	
53	\$all_negative = 0;
54	\$all_positive = 0;
55	
56	foreach (\$term_frequency2 as \$term) {
57	\$term = explode(" ", \$term);
58	
59	if (\$term[0]==\$kata_unik && \$term[1]=="1" && \$term[2]=="positif") {
60	\$total_positive=\$total_positive+1;
61	}

62	elseif (\$term[0]==\$kata_unik && \$term[1]=="1" && \$term[2]=="negatif") {
63	\$total_negative=\$total_negative+1;
64	}
65	}
66	
67	if (\$info_gain>=\$threshold) {
68	\$likelihood_positive = floatval((\$total_positive+1))/floatval((\$total_semua_positive+\$jumlah_kata_unik_baru));
69	\$likelihood_negative = floatval((\$total_negative+1))/floatval((\$total_semua_negative+\$jumlah_kata_unik_baru));
70	\$row1[] = \$likelihood_positive." ".\$kata_unik;
71	\$row2[] = \$likelihood_negative." ".\$kata_unik;
72	}
73	Else
74	{
75	\$likelihood_positive = floatval((0+1))/floatval((\$total_semua_positive+\$jumlah_kata_unik_baru));
76	\$likelihood_negative = floatval((0+1))/floatval((\$total_semua_negative+\$jumlah_kata_unik_baru));
77	\$row1[] = \$likelihood_positive." ".\$kata_unik;
78	\$row2[] = \$likelihood_negative." ".\$kata_unik;
79	}
80	}
81	
82	\$likelihood['positif'] = \$row1;
83	\$likelihood['negatif'] = \$row2;
84	
85	echo json_encode(\$likelihood);
86	}
87	}

Sedangkan untuk kelas view nya masih pada text_procesing.php dengan fungsi likelihood(info_gain,data,term_frequency2).

Kode Program 5.18 processing.php – likelihood (info_gain, data, term_frequency2)

1	
2	function likelihood(info_gain, data, term_frequency2)
3	{
4	th = document.getElementById("nilai-threshold").value;
5	\$.ajax({
6	url : "<?php echo site_url('Naive_bayes/likelihood')?>",
7	type: "POST",
8	data: {'data': data['words'], 'kata_unik_baru': info_gain, 'term_frequency2': term_frequency2, 'threshold': th},
9	dataType: "JSON",
10	progress: function(e) {
11	//menghitung panjang data
12	if(e.lengthComputable) {
13	//kalkulasi persentase terload
14	var pct = (e.loaded / e.total) * 100;
15	
16	//kunci persentase terload
17	
18	\$(".pengambilan-kata-unik").html("Sedang mengambil data..." + pct);
19	}
22	//this usually happens when Content-Length isn't set
21	else {
22	console.warn('Content Length not reported!');
23	}
24	},
25	success: function(kata_unik_baru)
26	{
27	\$(".pengambilan-entropy-partisi").html("Perhitungan entropy partisi sukses <i class='fa fa-check'></i>");
28	\$(".btn-generate-mulai").html('<i class="fa fa-spinner fa-spin"></i> menghitung nilai entropy');
29	

30	},
31	error: function (jqXHR, textStatus, errorThrown)
32	{
33	alert("error brooo");
34	}
35	
36	}); }

Proses penyimpanan nilai likelihood sebagai berikut :

Kode Program 5.19 Naive_bayes.php - simpan_likelihood()

1	//fungsi untuk menyimpan nilai likelihood setelah menerima //instruksi dari kelas controller, data yang diproses dari variabel likelihood dan id
2	function simpan_likelihood()
3	{
4	\$likelihood = \$this->input->post("likelihood");
5	\$id = \$this->input->post("id");
6	\$likelihood = explode(" ", \$likelihood);
7	\$data=array(//menyimpan nilai likelihood dalam bentuk array
8	'id' => \$id,
9	'kata' => \$likelihood[1],
10	'likelihood_positif' => \$likelihood[0],
11	'likelihood_negatif' => \$likelihood[2]
12	
13);
16	\$this->M_naivebayes->simpan_likelihood(\$data); //memanggil fungsi simpan_likelihood dari kelas M_naive bayes, supaya hasil yang diperoleh di simpan dalam tabel likelihood
15	}

Potongan kode kelas view pada fungsi untuk menyimpan likelihood sebagai berikut :

Kode Program 5.20 processing - simpan_likelihood(likelihood, id)

1	//fungsi untuk menyimpan nilai likelihood setelah menerima
2	//instruksi dari kelas controller, data yang diproses dari variabel likelihood dan id
3	function simpan_likelihood(likelihood, id)
4	{
5	\$('.btn-generate-mulai').html('<i class="fa fa-spinner fa-spin"></i> menyimpan likelihood...');
6	\$.ajax({
7	url : "<?php echo site_url('Naive_bayes/simpan_likelihood')?>",
8	type: "POST",
9	data: {'likelihood': likelihood, 'id': id},
10	dataType: "JSON",
11	progress: function(e) {
12	//menghitung panjang data
13	if(e.lengthComputable) {
14	//kalkulasi persentase terload
15	var pct = (e.loaded / e.total) * 100;
16	
17	//kunci persentase terload
18	
19	\$(".pengambilan-kata-unik").html("Sedang mengambil data..." + pct);
20	
21	}
22	//jika Content-Length tidak terset, maka
23	else {
24	console.warn('Content Length not reported!');
25	}
26	},
27	success: function(likelihood)
28	{
29	\$(".pengambilan-entropy-partisi").html("Perhitungan entropy partisi sukses <i class='fa fa-check'></i>");
32	},
33	error: function (jqXHR, textStatus, errorThrown)

34	{
35	
36	alert("error brooo");
37	}
38	}); }

Setelah dilakukan pemanggilan oleh kelas view terhadap kelas controller Naive_bayes.php, kelas controller akan meneruskan penyimpanan melalui kelas M_naivebayes.php, kelas inilah yang bertugas paling utama untuk menyimpan nilai-nilai likelihood yang didapat oleh program kedalam database.

Kode Program 5.21 M_naivebayes.php – simpan_likelihood(\$data)

1	<?php if (! defined('BASEPATH')) exit('No direct script access allowed');
2	
3	class M_naivebayes extends CI_Model {
4	
5	var \$tabel = "likelihood";
6	var \$tabel2 = "data_training";
7	var \$tabel3 = "prior";
8	var \$column = array("kata", "likelihood_positif", "likelihood_negatif");
9	//fungsi-fungs untuk mengolah data nilai likelihood, prior, dan probabilitas data latih maupun data uji kedalam dabatase sesuai dengan tablelnya
10	function simpan_likelihood(\$data)
11	{ //fungsi untuk menyimpan nilai likelihood pada tabel likelihood
12	\$this->db->insert(\$this->tabel, \$data);
13	}

Proses yang ketiga, mencari nilai prior yang didapat dengan membagi nilai dokumen yang berkelas tertentu dengan seluruh jumlah total dokumen pada data training. Fungsi yang digunakan adalah function hapus_likelihood_get_prior (), kurang lebihnya dapat di lihat pada source code berikut :

Kode Program 5.22 Naive_bayes.php - hapus_likelihood_get_prior()

1	
2	//fungsi untuk menghapus nilai likelihood sebelumnya dan menyimpan niai prior yang baru
3	function hapus_likelihood_get_prior()

4	{
5	\$this->M_naivebayes->hapus_likelihood();
6	\$positif = \$this->M_naivebayes->kalimat_positif();
7	\$negatif = \$this->M_naivebayes->kalimat_negatif();
8	//nilai prior didapat dengan membagi nilai \$positif (jumlah kalimat positif dlm database) dengan seluruh total dokumen di database)
9	\$prior_p = \$positif/(\$positif+\$negatif);
10	\$prior_n = \$negatif/(\$positif+\$negatif);
11	
12	\$data = array(
13	'prior_positif' => \$prior_p,
14	'prior_negatif' => \$prior_n,);
15	\$this->M_naivebayes->simpan_prior(\$data); //memanggil fungsi simpan_prior dari kelas M_naive bayes, supaya hasil yang diperoleh di simpan dalam tabel prior
16	}

Sedangkan pada kelas model proses yang dilakukan adalah memasukan sekaligus mengupdate nilai prior baru pada database.

Kode Program 5.23 M_naivebayes - get_prior()

1	
2	
3	function hapus_likelihood_get_prior(\$data) //fungsi untuk memasukan / mengisi nilai prior pada tabel
4	{
5	\$this->db->insert(\$this->tabel3, \$data);
6	}
7	
8	function hapus_likelihood()//fungsi untuk menghapus nilai likelihood / mengkosongkan tabel likelihood
9	{
10	\$this->db->empty_table("likelihood");
11	}
12	//menghitung jumlah kalimat positif
13	function kalimat_positif()

14	{
15	\$this->db->from(\$this->tabel2);
16	\$this->db->where("status", "positif");
17	return \$this->db->get()->num_rows();
18	}
19	//menghitung jumlah kalimat negatif
20	function kalimat_negatif()
21	{
22	\$this->db->from(\$this->tabel2);
23	\$this->db->where("status", "negatif");
24	return \$this->db->get()->num_rows();
25	}
26	
27	function get_likelihood(\$kata)
28	
29	\$this->db->from('likelihood');
30	\$this->db->where('kata', \$kata);
31	
32	return \$this->db->get()->row();
33	
34	//fungsi untuk mendapatkan nilai prior pada database
35	function get_prior()
36	{
37	\$this->db->from('prior');
38	
39	return \$this->db->get()->row(); }

Proses ke empat adalah mengkalikan nilai likelihood dan prior untuk mencari probabilitas kelas positif dan probabilitas kelas negatif pada tiap dokumen. Proses ini dapat dilihat pada subab berikut ini.

5.1.6 Implementasi Testing

Untuk implementasi Testing, dilakukan dengan memanggil kelas Tester.php, untuk mengontrol pemrosesan data test dari teks processing hingga pencarian

nilai probabilitasnya. Adapun kelas view untuk menampilkan data kelas data_tester.php.

Kode Program 5.24 Tester.php

1	<?php
2	defined('BASEPATH') OR exit('No direct script access allowed');
3	
4	class Tester extends CI_Controller {
5	
6	function __construct(){
7	parent::__construct();
8	\$this->load->model('M_processing');
9	\$this->load->model('M_naivebayes');
10	}
11	//fungsi testing digunakan untuk mendapatkan hasil preprocessing dari data uji
12	function testing()
13	{
14	\$words = \$this->input->post('word_tests'); //variabel \$words di inisiasikan dengan keyword 'word_test' yang berisikan dokumen uji yang baru dimasukkan
15	\$words = \$this->M_processing->clear_tag(\$words);//dilakukan proses clear_tag yang dipanggil dari kelas M_processing
16	\$this->load->library('Stemming_sastrawi');
17	\$stemmerFactory = new \Sastrawi\Stemmer\StemmerFactory();
18	\$stemmer = \$stemmerFactory->createStemmer();
19	
20	\$stemmed = \$stemmer->stem(\$words);
21	\$words = \$this->M_processing->stop_removal(\$stemmed); //dilakukan proses stop_removal yang dipanggil dari kelas M_processing
22	
23	echo json_encode(\$words);
24	}
25	//fungsi untuk mendapat nilai likelihood dan prior
26	function get_likelihood_dan_prior()
27	{
28	\$words = \$this->input->post('word_tests');

29	\$arraywords = explode(" ", \$words);
30	\$arraywords = array_filter(\$arraywords);
31	
32	\$likelihood = array();
33	
34	\$arrayKata = array();
35	foreach (\$arraywords as \$key) {
36	\$data = \$this->M_naivebayes->get_likelihood(\$key); //jika ditemukan kata pada dokumen uji yang sesuai dengan database kata unik beserta nilai likelihood nya, maka kata pada dokumen uji akan diberikan nilainya yang sama.
37	if (!empty(\$data)) {
38	\$likelihood[] = \$data->kata." ".\$data->likelihood_positif." ".\$data->likelihood_negatif;
39	}
40	
41	}
42	if (!empty(\$likelihood)) {
43	
44	\$arrayKata['likelihood'] = \$likelihood;
45	\$arrayKata['prior'] = \$this->M_naivebayes->get_prior(\$key);
46	}
47	Else
48	{
49	\$arrayKata['likelihood'] = "kosong"; //jika tidak ditemukan kata yang ada dalam database sehingga likelihood tidak nilainya, maka \$arraykata likelihood = kosong
50	\$arrayKata['prior'] = \$this->M_naivebayes->get_prior(\$key); //dilakukam proses get_prior yang dipanggil dari kelas M_naivebayes
51	}
52	
53	echo json_encode(\$arrayKata);
54	}
55	//menghitung nilai probabilitas pada data uji
56	function hitung_probablity()
57	{

58	\$data_prob = \$this->input->post('data_prob');
59	\$likelihood = \$data_prob['likelihood'];
60	if (\$likelihood=="kosong") { //jika nilai daripada \$likelihood = kosong, maka
61	\$probability['positif'] = "Kosong";
62	\$probability['negatif'] = "Kosong";
63	\$probability['hasil'] = "Mohon maaf kami <h2 style='color: red;'>tidak menemukan</h2> kata-kata anda pada data latih kami!";
64	}
65	Else
66	{
67	\$prior = \$data_prob['prior'];
68	\$prob_positif = 1;
69	\$prob_negatif = 1;
70	
71	foreach (\$likelihood as \$key) {
72	\$prob = explode(" ", \$key);
73	\$prob_negatif = \$prob[2]*\$prob_negatif;
74	
75	\$prob_positif = \$prob[1]*\$prob_positif;
76	}
77	
78	\$prob_positif = \$prob_positif*\$prior['prior_positif'];
79	\$prob_negatif = \$prob_negatif*\$prior['prior_negatif'];
80	
81	\$probability['positif'] = \$prob_positif;
82	\$probability['negatif'] = \$prob_negatif;
83	//menentukan hasil klasifikasi berdasarkan besaran nilai probabilitas
84	if (\$probability['positif']>\$probability['negatif']) { //jika hasil yang didapatkan dari variabel \$positif lebih besar dari pada \$negatif, maka
85	\$probability['hasil'] = "Kata-kata yang dimasukkan tergolong kategori <h2 style='color: blue;'>positif</h2>";
86	}
87	
88	Else //jika \$positif lebih kecil dari pada \$negatif, maka

89	{
90	\$probability['hasil'] = "Kata-kata yang dimasukkan tergolong kategori <h2 style='color: red;'>negatif</h2>";
91	}
92	}
93	
94	echo json_encode(\$probability);
95	}}

Sedangkan untuk potongan kode kelas view untuk proses testing ini sebagai berikut :

Kode Program 5.25 data_tester.php

1	
2	
3	url : "<?php echo site_url('Tester/testing')?>",
4	type: "POST",
5	data: {'word_tests': data},
6	dataType: "JSON",
7	progress: function(e) {
8	//menghitung panjang data
9	if(e.lengthComputable) {
10	//kalkulasi persentase terload
11	var pct = (e.loaded / e.total) * 100;
12	
13	//kunci persentase terload
14	\$(".proses-data").html("Sedang mengambil data..." + pct);
15	}
16	//jika Content-Length tidak terset, maka
17	else {
18	console.warn('Content Length not reported!');
19	}
20	},
21	success: function(data)
22	{

23	get_likelihood_dan_prior(data);
24	
25	},
26	error: function (jqXHR, textStatus, errorThrown)
27	{
28	alert('Error get data from ajax');
29	}
30	});
31	});
32	//fungsi untuk mendapat nilai likelihood dan prior setelah menerima //instruksi dari kelas controller, data yang diproses dari variabel word tests
33	function get_likelihood_dan_prior(data)
34	{
35	\$.ajax({
36	url : "<?php echo site_url('Tester/get_likelihood_dan_prior')?>",
37	type: "POST",
38	data: {'word_tests': data},
39	dataType: "JSON",
40	progress: function(e) {
41	//menghitung panjang data
42	if(e.lengthComputable) {
43	//kalkulasi persentase terload
44	var pct = (e.loaded / e.total) * 100;
45	
46	//kunci persentase terload
47	\$(".proses-data").html("Sedang mengambil data..." + pct);
48	}
49	//jika Content-Length tidak terset, maka
50	else {
51	console.warn('Content Length not reported!');
52	}
53	},
54	success: function(data_prob)

55	{
56	hitung_probablity(data_prob);
57	},
58	error: function (jqXHR, textStatus, errorThrown)
59	{
60	alert('Error get data from ajax');
61	}
62	});
63	}
64	//fungsi untuk mendapat nilai probabilitas setelah menerima //instruksi dari kelas controller, data yang diproses dari variabel data_prob
65	function hitung_probablity(data_prob)
66	{
67	\$.ajax({
68	url : "<?php echo site_url('Tester/hitung_probablity')?>",
69	type: "POST",
70	data: {'data_prob': data_prob},
71	dataType: "JSON",
72	progress: function(e) {
73	//menghitung panjang data
74	if(e.lengthComputable) {
75	//kalkulasi persentase terload
76	var pct = (e.loaded / e.total) * 100;
77	
78	//kunci persentase terload
79	\$(".proses-data").html("Sedang mengambil data..." + pct);
80	}
81	//jika Content-Length tidak terset
82	else {
83	console.warn('Content Length not reported!');
84	}
85	},
86	success: function(data_prob)

87	{
88	\$("#hasil").show();
89	\$("#probabilitas").show();
90	
91	\$("#probabilitas_positif").html(data_prob['positif']);
92	\$("#probabilitas_negatif").html(data_prob['negatif']);
93	
94	\$("#hasil").html(data_prob['hasil']);
95	},
96	error: function (jqXHR, textStatus, errorThrown)
97	{
98	alert('Error get data from ajax');
99	}
100	});
101	}
102	</script>

5.1.7 Implementasi Testing Data untuk 10 Fold Cross Validation

Implementasi Testing Data dibuat untuk dilakukan validasi terhadap data yang sudah dikumpulkan, data uji di masukan kedalam tabel uji dan data latih dimasukkan kedalam tabel data training, sedangkan hasilnya nanti muncul pada tabel tester_hasil. Berikut kelas controller Pengujian.php yang bertugas *handle* proses validasi.

Kode Program 5.26 Pengujian.php

1	?php
2	defined('BASEPATH') OR exit('No direct script access allowed');
3	
4	class Pengujian extends CI_Controller {
5	//proses fungsi - fungsi yang ada pada kelas ini, semuanya sama dengan fungsi yang sudah di bahas di beberapa kelas dalam proses processing, information gain, da naive bayes, hanya saja ini dalam lingkup pengujian (kelas controller Pengujian.php)
6	function __construct(){
7	parent::__construct();//untuk memproses kelas Pengujian.php diaman membutuhkan fungsi yang dapat meakukan proses processing, information

	gain, da naive bayes, maka
8	<code>\$this->load->model('M_processing');</code> //meload kelas model M_processing
9	<code>\$this->load->model('M_naivebayes');</code> //meload kelas model M_naive bayes
10	<code>\$this->load->model('M_pengujian');</code> //meload kelas model M_pengujian untuk mendapatkan nilai dari data uji yang sudah diklasifikasi dan menyimpannya dalam database
11	<code>}</code>
12	
13	<code>function get_data_uji()</code>
14	<code>{</code>
15	<code>\$data = \$this->M_pengujian->get_data_uji();</code>
16	
17	<code>echo json_encode(\$data);</code>
18	
19	<code>}</code>
20	//fungsi testing digunakan untuk mendapatkan hasil preprocessing dari data uji untuk proses 10 fold cross validation
21	<code>function testing()</code>
22	<code>{</code>
23	<code>\$words = \$this->input->post('word_tests');</code>
24	<code>\$data = array();</code>
25	<code>foreach (\$words as \$key) {</code>
26	
27	<code>\$words = \$this->M_processing->clear_tag(\$key['kalimat']);</code>
28	
29	<code>\$this->load->library('Stemming_sastrawi');</code>
30	<code>\$stemmerFactory = new \Sastrawi\Stemmer\StemmerFactory();</code>
31	<code>\$stemmer = \$stemmerFactory->createStemmer();</code>
32	
33	
34	
35	<code>\$stemmed = \$stemmer->stem(\$words);</code>
36	<code>\$words = \$this->M_processing->stop_removal(\$stemmed);</code>
37	

38	\$row['id'] = \$key['id'];
39	\$row['kalimat'] = \$words;
40	\$row['status'] = "";
41	\$data[] = \$row;
42	
43	}
44	
45	echo json_encode(\$data);
46	}
47	//fungsi untuk mendapat nilai likelihood dan prior untuk proses 10 fold cross validation
48	function get_likelihood_dan_prior()
49	{
50	\$words = \$this->input->post('word_tests');
51	\$data_all = array();
52	foreach (\$words as \$row) {
53	
54	
55	\$arraywords = explode(" ", \$row['kalimat']);
56	\$arraywords = array_filter(\$arraywords);
57	
58	\$likelihood = array();
59	
60	\$arrayKata = array();
61	foreach (\$arraywords as \$key) {
62	\$data = \$this->M_naivebayes->get_likelihood(\$key);
63	if (!empty(\$data)) {
64	\$likelihood[] = \$data->kata." ".\$data->likelihood_positif." ".\$data->likelihood_negatif;
65	}
66	}
67	if (!empty(\$likelihood)) {
68	
69	\$arrayKata['likelihood'] = \$likelihood;

70	\$arrayKata['prior'] = \$this->M_naivebayes->get_prior(\$key);
71	}
72	Else
73	{
74	\$arrayKata['likelihood'] = "kosong";
75	\$arrayKata['prior'] = \$this->M_naivebayes->get_prior(\$key);
76	}
77	
78	\$arrayKata['kalimat'] = \$row['kalimat'];
79	
80	\$row['id'] = \$row['id'];
81	\$row['kalimat'] = \$arrayKata;
82	\$row['status'] = "";
83	\$data_all[] = \$row;
84	}
85	
86	echo json_encode(\$data_all);
87	
88	}
89	//menghitung nilai probabilitas pada data uji untuk proses 10 fold cross validation
90	function hitung_probablity()
91	{
92	\$data_prob = \$this->input->post('data_prob');
93	foreach (\$data_prob as \$row) {
94	
95	\$likelihood = \$row['kalimat']['likelihood'];
96	if (\$likelihood=="kosong") {
97	\$simpan = array(
98	'id' => \$row['id'],
99	'kalimat' => \$row['kalimat']['kalimat'],
100	'status' => "Tidak Ditemukan",
101);
102	\$this->M_pengujian->simpan(\$simpan);

103	}
104	
105	Else
106	{
107	\$prior = \$row['kalimat']['prior'];
108	\$prob_positif = 1;
109	\$prob_negatif = 1;
110	
111	foreach (\$likelihood as \$key) {
112	\$prob = explode(" ", \$key);
113	\$prob_negatif = \$prob[2]*\$prob_negatif;
114	
115	\$prob_positif = \$prob[1]*\$prob_positif;
116	}
117	
118	\$prob_positif = \$prob_positif*\$prior['prior_positif'];
119	\$prob_negatif = \$prob_negatif*\$prior['prior_negatif'];
120	
121	\$probability['positif'] = \$prob_positif;
122	\$probability['negatif'] = \$prob_negatif;
123	
124	\$positif = floatval(\$probability['positif']);
135	\$negatif = floatval(\$probability['negatif']);
136	
137	if (\$positif>\$negatif) {
138	\$simpan = array(
139	'id' => \$row['id'],
140	'kalimat' => \$row['kalimat']['kalimat'],
141	'status' => "positif",
142);
143	\$this->M_pengujian->simpan(\$simpan);
144	// \$probability['hasil'] = "Kata-kata yang dimasukkan tergolong kategori <h2 style='color: red;'>positif</h2>";
145	}

146	
147	Else
148	{
149	\$simpan = array(
150	'id' => \$row['id'],
151	'kalimat' => \$row['kalimat']['kalimat'],
152	'status' => "negatif",
153);
154	\$this->M_pengujian->simpan(\$simpan);
155	// \$probability['hasil'] = "Kata-kata yang dimasukkan tergolong kategori <h2 style='color: red;'>negatif</h2>";
156	}
157	}
158	}
159	\$json = "berhasil";
160	echo json_encode(\$json);
161	}}

Sedangkan untuk kelas Modelnya M_pengujian.php bertugas meng-*handle* proses masuk dan keluarnya data pada database.

Kode Program 5.27 M_pengujian.php

1	<?php if (! defined('BASEPATH')) exit('No direct script access allowed');
2	
3	class M_pengujian extends CI_Model {
4	//fungsi untuk memanggil data uji pada database tabel "tester_uji"
5	function get_data_uji()
6	{
7	\$this->db->empty_table("tester_hasil");
9	
10	\$this->db->from('tester_uji');
11	
12	return \$this->db->get()->result_array();
13	}
14	//fungsi untuk menyimpan hasil kalsifikasi pada data uji kedalam

	database tabel "tester_hasil"
15	function simpan(\$data)
16	{
17	\$this->db->insert('tester_hasil', \$data);
18	}}

Untuk kelas view nya sebaga berikut :

Kode Program 5.28 pengujian.php

1	<script type="text/javascript">
2	\$(".btn-generettest").click(function()
3	{
4	\$.ajax({
5	url : "<?php echo site_url('Pengujian/get_data_uji')?>",
6	type: "GET",
7	// data: {'word_tests': data},
8	dataType: "JSON",
9	success: function(data) //saat proses yang dilakukan fungsi diatas berhasil, maka proses selanjutnya adalah meneruskan data untuk diolah oleh fungsi testing_processing
10	{
11	testing_processing(data);
13	},
14	error: function (jqXHR, textStatus, errorThrown)
15	{
16	alert('Error get data from ajax'); //jika proses gagal, maka muncul notifikasi error
17	}
18	});
19	});
20	//fungsi untuk melewati data yang akan maupun telah dilakukan proses preprocessing setelah menerima //instruksi dari kelas controller, data yang diproses dari variabel data
21	function testing_processing(data)
22	{

23	<code>\$.ajax({</code>
24	<code>url : "<?php echo site_url('Pengujian/testing')?>", //pemanggilan fungsi testing pada kelas controller Pengujian</code>
25	<code>type: "POST",</code>
26	<code>data: {'word_tests': data},</code>
27	<code>dataType: "JSON",</code>
28	<code>success: function(data)</code>
29	<code>{</code>
30	<code>get_likelihoood_dan_prior_pengujian(data); //jika proses berhasil, maka hasil olahan data akan diteruskan ke fungsi get_likelihoood_dan_pror_pengujian</code>
31	
32	<code>},</code>
33	<code>error: function (jqXHR, textStatus, errorThrown)</code>
34	<code>{</code>
35	<code>alert('Error get data from ajax');</code>
36	<code>}</code>
37	<code>});</code>
38	<code>}</code>
39	<code>//fungsi untuk melewati data yang akan dan telah dilakukan proses mencari nilai likelihoood dan prior setelah menerima //instruksi dari kelas controller, data yang diproses dari variabel data</code>
40	<code>function get_likelihoood_dan_prior_pengujian(data)</code>
41	<code>{</code>
42	<code>\$.ajax({</code>
43	<code>url : "<?php echo site_url('Pengujian/get_likelihoood_dan_prior')?>",</code>
44	<code>type: "POST",</code>
45	<code>data: {'word_tests': data},</code>
46	<code>dataType: "JSON",</code>
47	
48	<code>success: function(data_prob)</code>
49	<code>{</code>
50	<code>hitung_probablity_pengujian(data_prob); //jika proses berhasil, maka hasil olahan data akan diteruskan ke fungsi</code>

	hitung_probability_pengujian
51	},
52	error: function (jqXHR, textStatus, errorThrown)
53	{
54	alert('Error get data from ajax');
55	}
56	});
57	}
58	
59	function hitung_probablity_pengujian(data_prob)
60	{
61	\$.ajax({
62	url : "<?php echo site_url('Pengujian/hitung_probablity')?>", //pemanggilan fungsi hitung_probability di kelas Pengujian
63	type: "POST",
64	data: {'data_prob': data_prob},
65	dataType: "JSON",
66	progress: function(e) {
67	//hitung panjang data
68	if(e.lengthComputable) {
69	//kalkulasi persentase terload
70	var pct = (e.loaded / e.total) * 100;
71	
72	//kunci persentase terload
73	\$(".proses-data").html("Sedang mengambil data..." + pct);
74	}
75	//jika Content-Length tidak terset, maka
76	else {
77	console.warn('Content Length not reported!');
78	}
79	},
80	success: function(data_prob)
81	{
82	


83	},
84	error: function (jqXHR, textStatus, errorThrown)
85	{
86	alert('Error get data from ajax');
87	}
88	});
89	}
90	</script>

5.2 Implementasi Antarmuka Sistem

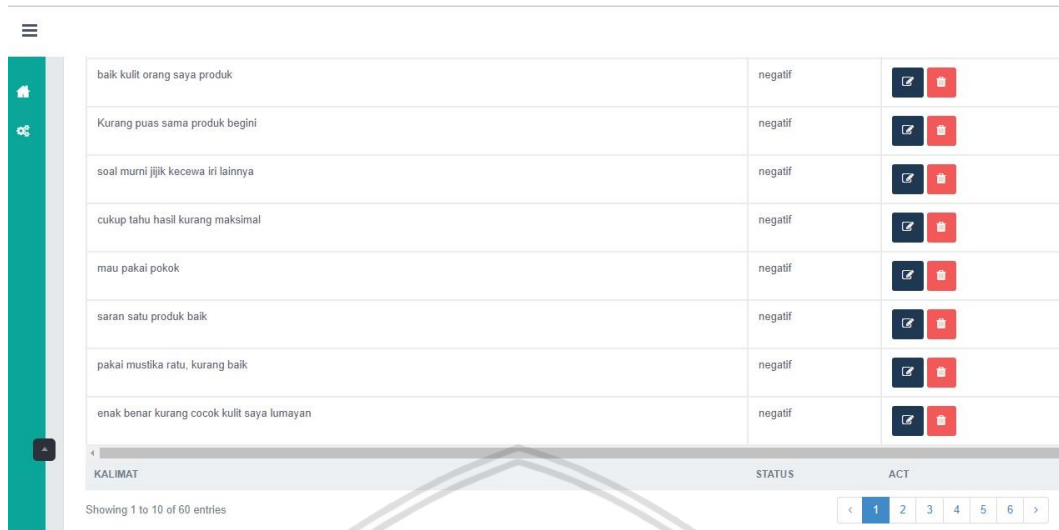
Implementasi antarmuka sistem ini menjelaskan hasil tampilan program yang sudah dirancang dan dijelaskan pada bab 3. Antarmuka pada bagian awal terdapat sebuah table yang berisi data training yang sudah dimasukkan oleh admin melalui database, dan di atasnya terdapat tombol untuk mengenerate data training untuk diolah supaya didapatkan hasil akhirnya.

















5.2.1 Interface Load Dokumen (Tampilan Awal)

Tampilan awal dari program ini saat dijalankan, sistem akan memproses load dokumen dan dimasukan kedalam tabel, yang nantinya akan di proses lebih lanjut saat user menekan tombol generate untuk memulai proses Preprocessing hingga pencarian nilai likelihhod tiap kata / fitur dan priornya.

KALIMAT	STATUS	ACT
mungkin produk pasar	negatif	 
biasa saya pakai produk aman pakai gatal gatal kulit	negatif	 
baik kulit orang saya produk	negatif	 

Gambar 5.1 Load Dokumen (Tampilan Awal)

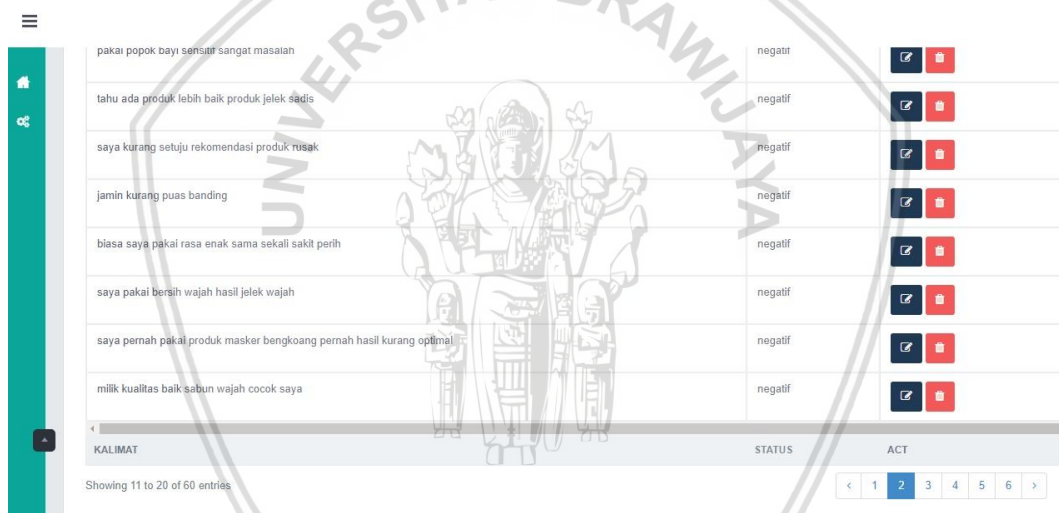


















baik kulit orang saya produk	negatif	 
Kurang puas sama produk begini	negatif	 
soal murni jujuk kecewa lri lainnya	negatif	 
cukup tahu hasil kurang maksimal	negatif	 
mau pakai pokok	negatif	 
saran satu produk baik	negatif	 
pakai mustika ratu, kurang baik	negatif	 
enak benar kurang cocok kulit saya lumayan	negatif	 
<div><div>KALIMAT</div><div>STATUS</div><div>ACT</div></div>		

Showing 1 to 10 of 60 entries

< 1 2 3 4 5 6 >

Gambar 5.2 Tampilan Tabel Dokumen bag. 1



pakai popok bayi sensitif sangat masalan	negatif	 
tahu ada produk lebih baik produk jelek sadis	negatif	 
saya kurang setuju rekomendasi produk rusak	negatif	 
jamin kurang puas banding	negatif	 
biasa saya pakai rasa enak sama sekali sakit perih	negatif	 
saya pakai bersih wajah hasil jelek wajah	negatif	 
saya pernah pakai produk masker bengkoang pernah hasil kurang optimal	negatif	 
milik kualitas baik sabun wajah cocok saya	negatif	 
<div><div>KALIMAT</div><div>STATUS</div><div>ACT</div></div>		

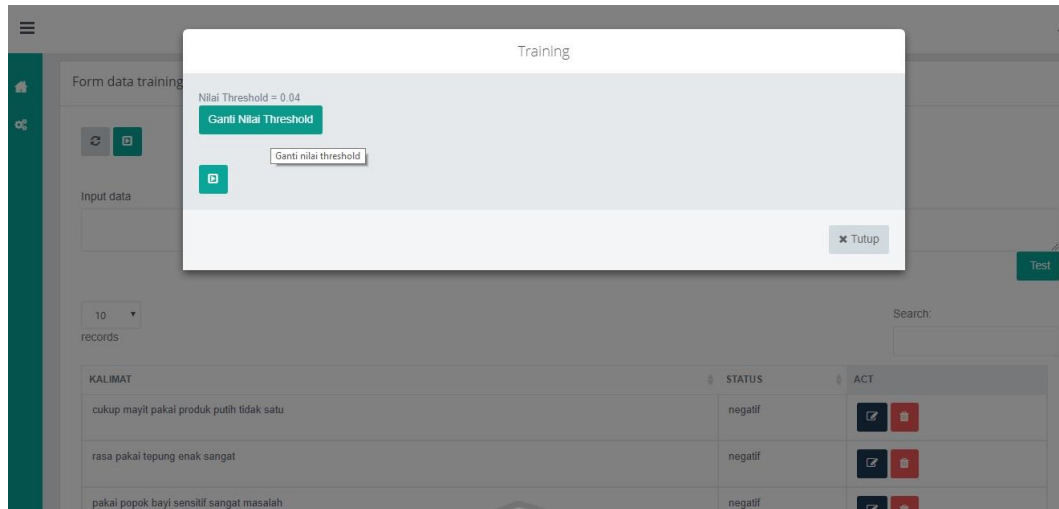
Showing 11 to 20 of 60 entries

< 1 2 3 4 5 6 >

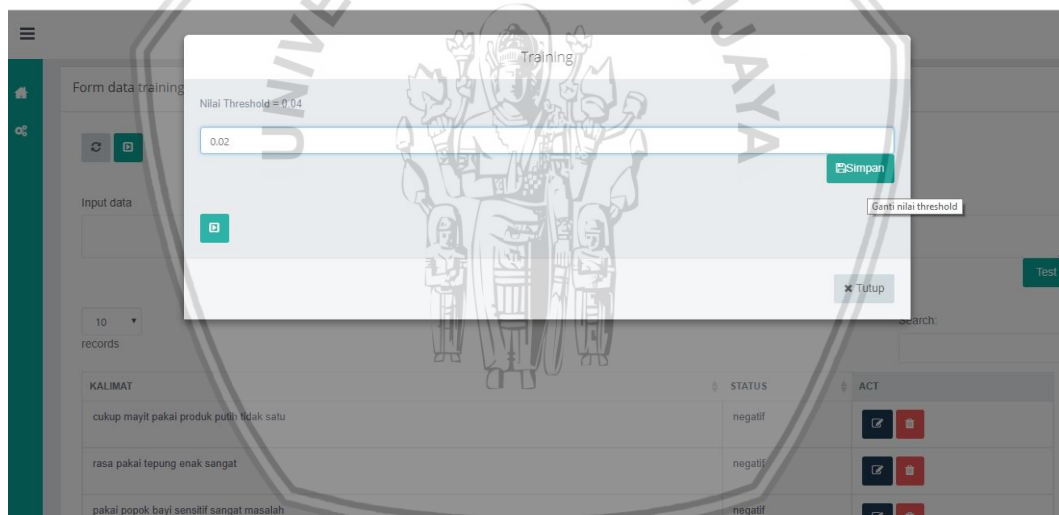
Gambar 5.3 Tampilan Tabel Dokumen bag.2

5.2.2 Interface Preprocessing

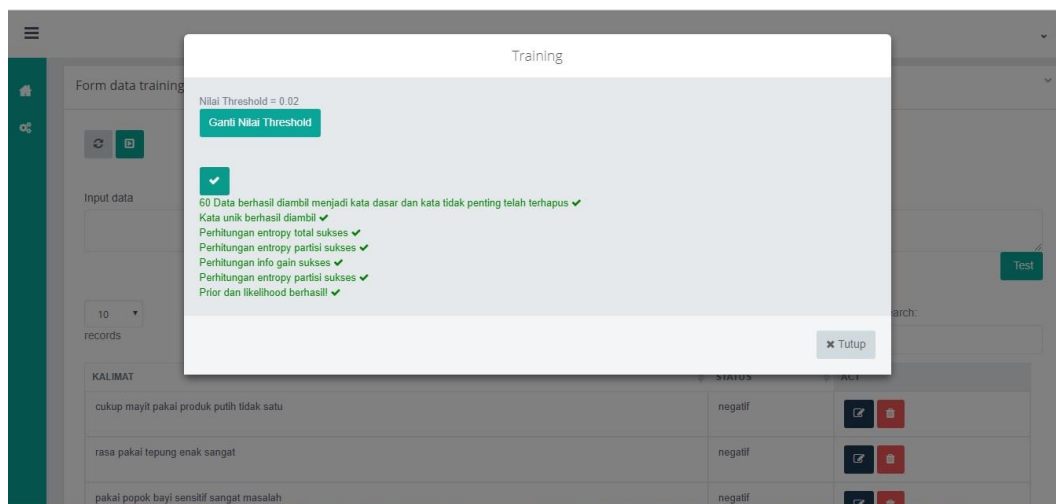
Pada bagian ini Interface tersebut akan muncul jika user telah menekan tombol generate, maka di tampilan akan muncul sebuah pop up untuk mengganti nilai threshold, kemudian setelah di simpan dan dijalankan program akan memproses dan tampilan yang dihasilkan adalah informasi terkait step-step langkah yang dilakukan oleh program, lebih jelasnya seperti Gambar di bawah ini.



Gambar 5.4 Tampilan Pop Up Tthreshold Awal



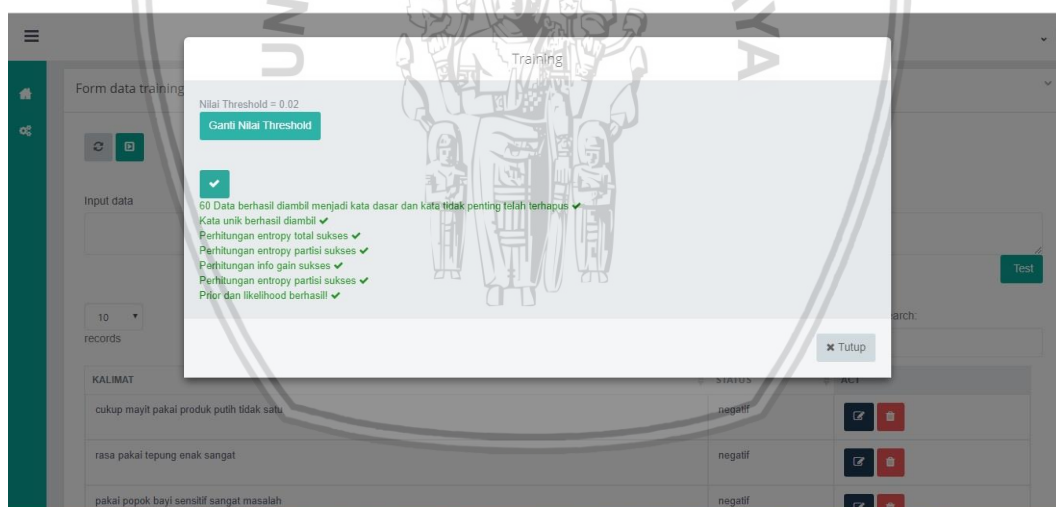
Gambar 5.5 Tampilan Pop Up Ubah Threshold



Gambar 5.6 Tampilan Pop Up Proses Training (*Preprocessing*)

5.2.3 Interface Klasifikasi

Tampilan interface Klasifikasi hampir sama dengan Interface Preprocessing, tampilannya menunjukkan step step proses klasifikasi, seperti pada Gambar di bawah ini.



Gambar 5.7 Tampilan Pop Up Proses Training (Kalsifikasi)

5.2.4 Interface Testing

Tampilan interface testing adalah menghasilkan nilai probabilitas dari kelas positif dan kelas negatif, dengan memasukkan sebuah data uji baru dan menekan tombol test, seperti pada Gambar berikut.

The screenshot shows a web application interface for testing. It has a teal sidebar on the left with a menu icon. The main area is titled 'Form data training'. Below the title are two buttons: a refresh icon and a 'Test' button. The 'Input data' field contains the text: 'lipstick #mustikaratu kalau di pakai lembut di bibir hasilnya bagus banget rekomendasi lah'. Below the input field, there are two boxes for probabilities: 'PROBABILITAS POSITIF' with the value '9.1296534062425e-11' and 'PROBABILITAS NEGATIF' with the value '3.0761019185339e-11'. A yellow box below these displays the result: 'Kata-kata yang dimasukkan tergolong kategori positif'. At the bottom, there is a dropdown menu set to '10' and a 'Search' field.

Gambar 5.8 Tampilan Testing Hasil Positif

The screenshot shows the same web application interface as Gambar 5.8, but with a different input and result. The 'Input data' field contains the text: 'slalan banget nih Cuma jany dan harapan palsu, hasil gak memuaskan'. The probability values are 'PROBABILITAS POSITIF' with the value '5.9261863471398e-25' and 'PROBABILITAS NEGATIF' with the value '1.2044901838205e-22'. The yellow box displays the result: 'Kata-kata yang dimasukkan tergolong kategori negatif'. The 'Test' button is now labeled 'Test-all'. The bottom of the interface shows a Windows taskbar with various icons and a system clock indicating 11:07 on 03/01/2018.

Gambar 5.9 Tampilan Testing Hasil Negatif

BAB 6 PENGUJIAN DAN ANALISIS

6.1 Pengujian

Sesuai dengan hasil manualisasi dan hasil sistem di bab sebelumnya, pada bagian ini menjelaskan analisis dan pembahasan dari pengujian atas hasil-hasil tersebut. Pengujian yang dilakukan adalah dengan melakukan pengujian *10-fold cross validation*, dan dicari nilai rata-rata akurasi yang terbaik dari tiap-tiap threshold yang digunakan.

6.1.1 Pengujian dan analisis validasi terhadap data dengan metode *10-fold cross validation*

Pengujian ini menggunakan metode *10-fold cross validation*, dimana seluruh data latih yang berjumlah 200 data dibagi menjadi 10 bagian (k1, k2, k3, k4, k5, k6, k7, k8, k9, k10), kemudian tiap satu bagian tersebut dijadikan data uji dan sisa sembilan bagian digunakan menjadi data latih, begitu seterusnya bergantian. Selanjutnya pada tiap pengujian data uji (k bagian) dilakukan pengfilteran dengan metode *Information Gain*, besaran tindakan pengfilteran berbanding lurus dengan besaran nilai threshold, nilai threshold yang dipakai berkisar antara lain : 0, 0.01, 0.02, 0.03, 0.04, 0.05, 0.06, 0.07, 0.08, 0.09, dan terakhir 0.10, semuanya dipakai pada tiap k bagian. Kemudian dicari nilai rata-rata akurasi terbaik dari setiap threshold yang dipakai pada masing-masing k bagian.

Nilai threshold diatas ditetapkan untuk pengujian ini atas dasar asumsi dan referensi dari penelitian terkait sebelumnya yang dilakukan oleh Betha Nurina Sari dengan judul Implementasi Teknik Seleksi Fitur *Information Gain* pada Algoritma Klasifikasi *Machine Learning* untuk Prediksi Performa Akademik Siswa, dimana nilai threshold 0 memberikan makna tidak digunakan metode seleksi fitur *Information Gain* pada proses sentimen analisis ini, dan tingkatan kenaikan threshold sebesar 0.01 untuk menandakan penggunaan metode *Information Gain*, sedangkan kenaikan yang terus meningkat hingga 0.10 adalah inisiatif peneliti untuk pengembangan pencarian nilai threshold terbaik untuk menfilter data yang digunakan pada penelitian ini, singkatnya semakin besar nilai threshold yang dipakai, maka tindakan penfilteran oleh metode seleksi fitur *Information Gain* pun semakin besar.

Berikut hasil pengujian ini dipaparkan dalam bentuk tabel berikut :

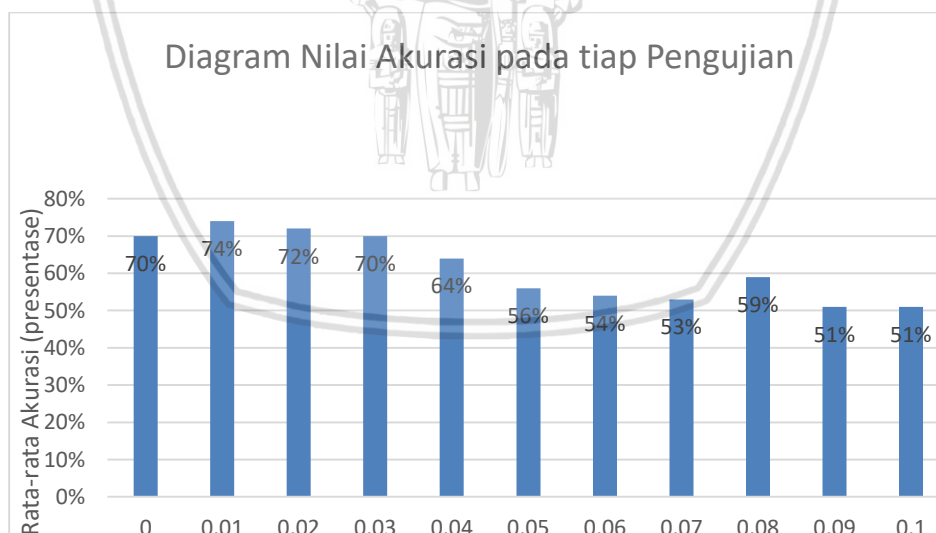
Tabel 6.1 Tabel Akurasi Tiap K bagian

		K Bagian (Tiap K berisi 20 Dokumen)										Rata-rata Akurasi
		k1	k2	k3	k4	k5	k6	k7	k8	k9	k10	
Threshold	0	60%	65%	60%	80%	70%	80%	75%	70%	65%	70%	70%
	0,01	75%	70%	60%	80%	85%	85%	75%	70%	70%	70%	74%
	0,02	85%	75%	65%	75%	65%	75%	85%	60%	65%	65%	72%

Threshold	0,03	75%	75%	65%	65%	60%	65%	70%	75%	80%	65%	70%
	0,04	50%	65%	70%	60%	60%	65%	65%	65%	75%	60%	64%
	0,05	50%	60%	55%	65%	55%	50%	55%	50%	60%	60%	56%
	0,06	50%	60%	55%	50%	55%	50%	50%	50%	60%	55%	54%
	0,07	50%	50%	55%	50%	55%	50%	50%	50%	60%	55%	53%
	0,08	50%	50%	50%	50%	55%	70%	85%	65%	50%	60%	59%
	0,09	50%	50%	50%	50%	55%	50%	50%	50%	50%	50%	51%
	0,1	50%	50%	50%	50%	55%	50%	50%	50%	50%	50%	51%

Sesuai dengan hasil tabel 6.1, nilai threshold yang terbaik dalam memberikan hasil akurasi yang optimal ada pada nilai threshold 0.01 dengan nilai rata-rata akurasi sebesar 74%, sedangkan nilai threshold 0 dengan nilai rata-rata akurasi sebesar 70%. Hal ini menjelaskan bahwa penambahan penggunaan metode seleksi fitur *Information Gain* pada proses analisis sentimen dapat mempengaruhi hasil yang lebih optimal dalam mengklasifikasikan dokumen pada kelas yang sesuai dengan selisih yang didapatkan 4% dibandingkan dengan tidak menambahkan metode seleksi fitur didalam proses analisis sentimen. Lebih lanjutnya dapat dilihat pada tabel 7.2 diagram batang yang menunjukkan perbandingan nilai akurasi yang didapat dari tiap-tiap pengujian.

Gambar 6.1 Diagram Nilai Akurasi pada tiap Pengujian



Selisih dengan 4% dikarenakan pada beberapa faktor diantaranya jumlah data latih yang kurang kompleks memicu nilai akurasi, karena semakin kompleks data latih yang ditandai dengan banyak jumlah dokumen yang digunakan dalam memproses akan menghasilkan nilai akurasi yang semakin meningkat dari sebelumnya. Hal ini dapat dibuktikan dengan analisis perbandingan jumlah data latih yang digunakan untuk menguji data uji sebanyak 10 dokumen (5 positif, 5 negatif) sebagai berikut :

Tabel 6.2 – Hasil klasifikasi dengan data latih 60 data

No	Dokumen	Pakar	Klasifikasi Program
1	Mama sampek afal, hobby ku beli produk mustika tiap minggu hahhahaaaa	Positif	Negatif
2	Nda mau pindah kelain hati	Positif	Negatif
3	Dulu gue, minum teh hijau mustika ratu , trs kalo makan banyak juga	Positif	Positif
4	Masker mustika ratu mantab:D	Positif	Positif
5	Pacar ndak mau aku pkek selain lotion ini	Positif	Negatif
6	Please jangan pakek, kecuali lu mau nambah jerawat	Negatif	Negatif
7	Sukaakkkk, suka bakar produknya wkwkwkwkk	Negatif	Positif
8	pengen garuk kulit terus, sialan	Negatif	Negatif
9	Kulit berasa kebakar, mual sama bau creamnya, ndak kuat !	Negatif	Negatif
10	Jangankan beli 10, 1 aja ogah	Negatif	Negatif
Akurasi			60%

Sedangkan jika menggunakan 120 data latih, maka hasil klasifikasi meningkat dengan akurasi 70%.

Tabel 6.3 – Hasil Klasifikasi dengan data latih 120 data

No	Dokumen	Pakar	Klasifikasi Program
1	Mama sampek afal, hobby ku beli produk mustika tiap minggu hahhahaaaa	Positif	Positif
2	Nda mau pindah kelain hati	Positif	Negatif
3	Dulu gue, minum teh hijau mustika ratu , trs kalo makan banyak juga	Positif	Positif
4	Masker mustika ratu mantab:D	Positif	Positif
5	Pacar ndak mau aku pkek selain lotion ini	Positif	Negatif

6	Please jangan pakek, kecuali lu mau nambah jerawat	Negatif	Negatif
7	Sukaakkkk, suka bakar produknya wkwkwkwkk	Negatif	Positif
8	pengen garuk kulit terus, sialan	Negatif	Negatif
9	Kulit berasa kebakar, mual sama bau creamnya, ndak kuat !	Negatif	Negatif
10	Jangankan beli 10, 1 aja ogah	Negatif	Negatif
Akurasi			70%

Pada hasil tabel 6.3, menjelaskan bahwa penggunaan dokumen yang lebih banyak akan memberikan peluang terhadap kompleksitas dan kelengkapan jenis kata yang di proses oleh sistem untuk tahap pelatihan. Hal lain yang menjadi faktor sebagai penyebab hasil akurasi pengujian pada penelitian ini kurang significant adalah penyebaran jenis kata yang tidak merata, atau dengan kata lain jumlah jenis kata tertentu pada data latih dokumen positif tidak sama / seimbang dengan jenis kata tertentu pada data latih dokumen negatif. Lebih jelasnya, pada K3 hasil klasifikasi dengan 0.01 adalah sebagai berikut :

Tabel 6.4 – Hasil Klasifikasi K3 dengan threshold 0.01

No.	Dokumen	Pakar	Sistem
			0,01
1	Kotak kosmetik dipenuhi cream ini	Positif	Positif
2	Nda mau pindah kelain hati	Positif	Negatif
3	Dulu gue, minum teh hijau mustika ratu , trs kalo makan banyak juga	Positif	Positif
4	Masker mustika ratu mantab:D	Positif	Positif
5	Pacar ndak mau aku pkek selain lotion ini	Positif	Negatif
6	Berdua sama pacar pakai ini, betah deh	Positif	Negatif
7	kurang tahu kalu produk lain kyk gimana efeknya, tapi mustika ratu yang terbaik	Positif	Negatif
8	Beli ilagi, beli terus !	Positif	Negatif
9	Buat rambut gk rontok sih bgus katanya abis itu pake shampoo nya merk mustika ratu juga	Positif	Positif
10	Mama sampek afal, hobby ku beli produk mustika tiap minggu hahhahaaaa	Positif	Positif
11	saya habis pake pembersih wajah, hasilnya jelek di Mukak	Negatif	Positif
12	Soalnya mustika ratu kan ga murni tuh, jadi jijik, truss kecewa iri dan lainnya	Negatif	Negatif
13	Alaaamakkk pedih muka gue,dijadiin mummy ama produk ini	Negatif	Negatif

14	Please jangan pakek, kecuali lu mau nambah jerawat	Negatif	Negatif
15	Sukaakkkk, suka bakar produknya wkwkwkkwkk	Negatif	Positif
16	pengen garuk kulit terus, sialan	Negatif	Negatif
17	Kulit berasa kebakar, mual sama bau creamnya, ndak kuat !	Negatif	Negatif
18	Jangan beli 10, 1 aja ogah	Negatif	Negatif
19	tidak baik untuk kulit orang seperti saya produk mustika ratu ini miskin !	Negatif	Negatif
20	Bukannya makin bersih, malah kusam banget sampek leher	Negatif	Positif
		Akurasi	60%

Sesuai dengan tabel 6.4 sebagai sampel dokumen 6 oleh pakar divalidasi bahwa dokumen tersebut bersentimen positif (masuk dalam kategori kelas positif), namun saat program dijalankan, yang terbaca adalah sebaliknya, dokumen 6 masuk pada kelas negatif. Sebeum di analisis lebih lanjut, pada dasarnya program akan menentukan nilai probabilitas suatu dokumen berdasarkan besaran nilai likelihood tiap kata dan berapa kali likelihood di kalikan sesuai banyak jumlah kata yang sama pada data uji terhadap data latih.

Pada dokumen 6, jenis kata dan jumlah kata yang tertera adalah sebagai berikut :

Tabel 6.5 – Jenis kata dan jumlah kata pada dokumen 6 K3

Jenis Kata	Jumlah kata positif (np)	Jumlah kata negatif (nn)
Dua	0	1
Pakai	11	11
Betah	0	1
Total	11	13

Total banyak jumlah kata negatif lebih banyak daripada kata positif, hal ini membuat total perkalian nilai likelihood negatif lebih besar daripada likelihood positif.

Namun saat threshold mencapai 0.10, dipastikan penggunaan Information gain cukup besar dan mempengaruhi penghilangan kata-kata pada dokumen 6, kemudian digantikan oleh *laplace smoothing* sehingga berapapun jumlah kata tertentu tetap bernilai 1. Sedangkan jika dilihat kondisi penggunaan *laplace smoothing* secara penuh (IG digunakan secara maksimal dengan threshold 1) pada data latih, nilai likelihood positif yang tersimpan pada database lebih besar daripada likelihood negatif.

Tabel 6.6 – Tabel nilai likelihood saat threshold Information Gain bernilai 1

No.	Kata	Likelihood Positif	Likelihood Negatif
1	Lebih	0.0009469696969697	0.00085763293310463
2	Baik	0.0009469696969697	0.00085763293310463
3	Alihh	0.0009469696969697	0.00085763293310463
4	Produk	0.0009469696969697	0.00085763293310463
5	Wajah	0.0009469696969697	0.00085763293310463

Hal ini dikarenakan jumlah kata dalam dokumen positif lebih sedikit daripada jumlah kata dalam dokumen negatif pada seluruh dokumen yang menjadi data latih. Dari penjelasan tersebut, hasil klasifikasi pada dokumen 6 K3 menjadi positif semula sesuai hasil validasi pakar. Namun tidak hanya pada dokumen 6 saja, bahkan untuk dokumen yang berkelas negatif (dokumen 11 – 20) berubah klasifikasinya menjadi positif semua karena ada pergeseran nilai likelihood yang menjadikan probabilitas positif pada masing-masing dokumen menjadi lebih besar dari sebelumnya.

Tabel 6.7 – Hasil Klasifikasi K3 dengan threshold 0.10

No.	Dokumen	Pakar	Sistem
			0,01
1	Kotak kosmetik dipenuhi cream ini	Positif	positif
2	Nda mau pindah kelain hati	Positif	positif
3	Dulu gue, minum teh hijau mustika ratu , trs kalo makan banyak juga	Positif	positif
4	Masker mustika ratu mantab:D	Positif	positif
5	Pacar ndak mau aku pkek selain lotion ini	Positif	positif
6	Berdua sama pacar pakai ini, betah deh	Positif	positif
7	kurang tahu kalu produk lain kyk gimana efeknya, tapi mustika ratu yang terbaik	Positif	positif
8	Beli ilagi, beli terus !	Positif	positif
9	Buat rambut gk rontok sih bgus katanya abis itu pake shampoo nya merk mustika ratu juga	Positif	positif
10	Mama sampek afal, hobby ku beli produk mustika tiap minggu hahhahaaaa	Positif	positif
11	saya habis pake pembersih wajah, hasilnya jelek di Mukak	Negatif	positif
12	Soalnya mustika ratu kan ga murni tuh, jadi jijik, truss kecewa iri dan lainnya	Negatif	positif

13	Alaaamakkkkk pedih muka gue,dijadiin mummy ama produk ini	negatif	Positif
14	Please jangan pakek, kecuali lu mau nambah jerawat	negatif	Positif
15	Sukaakkkk, suka bakar produknya wkwkwkwkk	negatif	Positif
16	pengen garuk kulit terus, sialan	negatif	Positif
17	Kulit berasa kebakar, mual sama bau creamnya, ndak kuat !	negatif	Positif
18	Jangankan beli 10, 1 aja ogah	negatif	Positif
19	tidak baik untuk kulit orang seperti saya produk mustika ratu ini miskin !	negatif	Positif
20	Bukannya makin bersih, malah kusam banget sampek leher	negatif	Positif
		Akurasi	50%

Dilihat secara keseluruhan pula, nilai rata-rata akurasi saat threshold mencapai 0.03 hingga 0.10 hasil yang didapat menjadi sama dengan saat penggunaan threshold 0 bahkan menjadi turun drastis hingga pada 51%, hal ini membuktikan tidak semua nilai threshold dapat digunakan untuk proses seleksi fitur dalam proses analisis sentimen, dikarenakan ada batasan penggunaan nilai threshold yang dipakai, dan batas maksimal nilai threshold ada pada 0.02 dengan nilai akurasi selisih 2% dengan nilai akurasi threshold 0 sebesar 72%.

Selain itu nilai akurasi yang paling tinggi didapatkan pada K1 (threshold 0.02), kemudian K5, K6 (threshold 0.01), dan K7 (threshold 0.02 dan 0,08) dengan presentase 85%, sedangkan pada k bagian tertentu dengan threshold tertentu program mengenali dokumen dalam kelas positif atau negatif pada titik terendah sebesar 50%. Sedangkan untuk representasi adanya nilai akurasi pada tiap kolom dan baris, menandakan bahwa pada setiap k bagian program dapat memproses setiap dokumen didalamnya tanpa ada yang terlewatkan. Hal tersebut dapat dibuktikan dengan menguji sistem dengan data latih dan data uji yang sama persis, dan hasilnya disajikan dalam bentuk tabel 7.7.

Tabel 6.8 Hasil Klasifikasi K3 dengan data latih dan data uji sama

No.	Dokumen	Pakar	Sistem
			0,01
1	Kotak kosmetik dipenuhi cream ini	positif	Positif
2	Nda mau pindah kelain hati	positif	Positif
3	Dulu gue, minum teh hijau mustika ratu , trs kalo makan banyak juga	positif	Positif
4	Masker mustika ratu mantab:D	positif	Positif
5	Pacar ndak mau aku pkek selain lotion ini	positif	Positif
6	Berdua sama pacar pakai ini, betah deh	positif	Positif

7	kurang tahu kalau produk lain kayak gimana efeknya, tapi mustika ratu yang terbaik	positif	Positif
8	Beli ilagi, beli terus !	positif	Positif
9	Buat rambut gak rontok sih bagus katanya abis itu pake shampoo nya merk mustika ratu juga	positif	Positif
10	Mama sampek afal, hobby ku beli produk mustika tiap minggu hahhahaaaa	positif	Positif
11	saya habis pake pembersih wajah, hasilnya jelek di Mukak	negatif	negatif
12	Soalnya mustika ratu kan ga murni tuh, jadi jijik, truss kecewa iri dan lainnya	negatif	negatif
13	Alaaamakkkkk pedih muka gue, dijadiin mummy ama produk ini	negatif	negatif
14	Please jangan pakek, kecuali lu mau nambah jerawat	negatif	negatif
15	Sukaakkkk, suka bakar produknya wkwkwkwkk	negatif	negatif
16	pengen garuk kulit terus, sialan	negatif	Negatif
17	Kulit berasa kebakar, mual sama bau creamnya, ndak kuat !	negatif	Negatif
18	Jangankan beli 10, 1 aja ogah	negatif	Negatif
19	tidak baik untuk kulit orang seperti saya produk mustika ratu ini miskin !	negatif	Negatif
20	Bukannya makin bersih, malah kusam banget sampek leher	negatif	Negatif
		Akurasi	100%

BAB 7 PENUTUP

7.1 Kesimpulan

Setelah dilakukan analisis, perancangan, dan pengujian sistem pada **Penerapan Analisis Sentimen untuk Menilai Suatu Produk pada Twitter Berbahasa Indonesia dengan Metode *Naïve Bayes Classifier* dan *Information Gain*** maka dapat diambil kesimpulannya sebagai berikut :

1. Penggunaan Metode *Naive Bayes* dipilih sebagai metode klasifikasi pada penelitian ini dikarenakan prosesnya yang sederhana dan *Information Gain* ditambahkan untuk mengoptimalkan hasil dari klasifikasi *Naive Bayes*, penyusunan alur algoritma proses analisis sentimen ini diawali dengan proses Preprocessing, kemudian dilanjut dengan proses *Information Gain* untuk menseleksi fitur / kata yang kurang relevan dan dianggap sebagai noise pada proses klasifikasi selanjutnya yang dilakukan oleh metode *Naive Bayes*.
2. Sedangkan data Twitter yang digunakan didapat dari proses pemanfaatan API twitter untuk mengambil data dari beberapa akun Twitter, dan hasil pengisian formulir online untuk menuliskan sentimen terhadap produk Mustika Ratu oleh rekan-rekan peneliti.
3. Dengan Jumlah data sebanyak 200 data (100 dokumen positif, dan 100 dokumen negatif) peningkatan akurasi yang didapatkan dengan selisih 4%, karena nilai rata-rata akurasi yang didapat jika tanpa menggunakan metode *Information Gain* (threshold 0) sebesar 70%, sedangkan dengan menggunakan metode *Information Gain* (threshold 0.01) hasil rata-rata akurasi yang didapatkan sebesar 74%.
4. Perolehan hasil tertinggi nilai rata-rata akurasi yg mencapai 74% didasari faktor seperti jumlah data latih yang kurang banyak, dimana penulis membuktikan melalui tabel 6.2 data uji sebanyak 10 dokumen mencapai akurasi 60% saat data latih yang digunakan 60 dokumen, sedangkan saat menggunakan data latih yang lebih banyak lagi sebanyak 120 dokumen (tabel 6.3) hasil akurasi meningkat menjadi 70%.
5. Adapun faktor lainnya, yaitu penyebaran jenis kata yang tidak merata atau dengan kata lain jumlah jenis kata tertentu pada data latih dokumen positif tidak sama / seimbang dengan jenis kata tertentu pada data latih dokumen negatif, sebagai sample dokumen 6 pada K3 memiliki total banyak jumlah kata negatif sebanyak 12 kata lebih banyak daripada kata positif sebanyak 11 kata, hal ini membuat total perkalian nilai likelihood negatif lebih besar daripada likelihood positif, sehingga dokumen 6 yang seharusnya berkelas positif oleh sistem di baca menjadi kelas negatif.
6. Jika dilihat kondisi penggunaan *laplace smoothing* secara penuh (IG digunakan secara maksimal dengan threshold 1) pada data latih untuk K3, nilai likelihood positif yang tersimpan pada database lebih besar daripada likelihood negatif sehingga hasil klasifikasi pada dokumen 6 K3 menjadi positif semula.. Namun pada dokumen berkelas negatif (dokumen 11 – 20)

berubah klasifikasinya menjadi positif semua karena ada pergeseran nilai likelihood yang menjadikan probabilitas positif pada masing-masing dokumen menjadi lebih besar dari sebelumnya.

7. Tidak semua nilai threshold dapat digunakan untuk proses seleksi fitur metode *Information Gain* dalam proses analisis sentimen, dikarenakan ada batasan penggunaan nilai threshold yang dipakai, dan batas maksimal nilai threshold ada pada 0.03 dengan nilai rata-rata akurasi 72%, selisih 2% jika dibandingkan dengan hasil rata-rata akurasi tanpa menggunakan metode *Information Gain*.
8. Nilai akurasi yang paling tinggi didapatkan pada K1 (threshold 0.02), kemudian K5, K6 (threshold 0.01), dan K7 (threshold 0.02 dan 0,08) dengan presentase 85%, sedangkan pada k bagian tertentu dengan threshold tertentu program mengenali dokumen dalam kelas positif atau negatif pada titik terendah sebesar 50%.
9. Kemudian representasi adanya nilai akurasi pada tiap kolom dan baris, menandakan bahwa pada setiap K bagian program dapat memproses setiap dokumen didalamnya tanpa ada yang terlewatkan.

7.2 Saran

Adapun saran dalam penelitian ini sebagai bentuk acuan dalam pengembangan lebih lanjut sebagai berikut :

1. Supaya mendapatkan hasil dengan akurasi yang lebih bagus lagi, dapat ditambahkan jumlah dokumen dengan jenis kata yang lebih kompleks lagi dan komposisi antara dokumen positif dan dokumen negatif yang relatif sama.
2. Metode *Naive Bayes* sebagai metode klasifikasi memiliki potensi untuk dikombinasikan dengan metode lainnya dalam menseleksi fitur, mengingat adanya kesederhanaan proses pada Metode *Naive Bayes*.
3. Bahasa pemrograman yang digunakan dapat dipilih yang lebih tepat atau penyusunan algoritma yang lebih ringkas supaya dapat memangkas waktu menjadi lebih singkat dari sebelumnya, hal ini dikarenakan terkadang dalam proses penggunaan sistem program dibutuhkan kesabaran agar terhindar dari overload runtime.

DAFTAR PUSTAKA

- Abdul Kadir. (2001), *Dasar Pemrograman Web Dinamis Menggunakan PHP*, C.V Andi Offset, Yogyakarta.
- Andi, Madcoms. 2010, *Adobe Dreamweaver CS5 dengan Pemrograman PHPMySQL*. Penerbit. CV Andi Offset.
- Andilala. "Movie Review Sentimen Analisis dengan Metode Naive Bayes on Feature Selection", *Jurnal Pseudocode*, Februari 2016, vol 3 (1), 3-5.
- Basuki, A. P., 2014. *Proyek Membangun Website Berbasis PHP dengan codeIgniter 1st ed*. Yogyakarta: CV Lokomedia
- Capece, D., person, K., Lyon, E. & Lane, T., 2009. *Digital Influence in News and Politics*. New York: Spaaarxoo.
- Dragut, E., Fang, F., Sistla, P. & Lane, T., 2009. Stop Word and Related problems in Web Interface Integration. *VLDBEndowment*, pp. 349-360.
- Feldman, R. & Sanger, J., 2006. *The Text Mining HandBook*. Cambridge: Cambridge University Press.
- Grafelly, Delvit Bagaimana perkembangan Twitter saat ini? 13 Maret 2017. <http://www.techno.id/social/bagaimana-perkembangan-twitter-saat-ini-1509122.html>.
- Hidayat, A. N., (2015). "Analisis Sentimen Terhadap Wacana Politik pada Media Masa Online Menggunakan Algoritma Support Vector Machine dan Naïve Bayes". *Jurnal Elektronik Sistem Informasi dan Komputer (JESIK)*. 1 (1), 1-2.
- Industri.kontan, 2017. *Mustika Ratu Mencetak Laba di Kuartal 3*. 20 Maret 2018. <https://industri.kontan.co.id/news/mustika-ratu-mencetak-laba-rp-15-miliar-di-q3>
- Tekno.Kompas, 2017. *Pengguna Twitter Naik 9 Juta, sejak 2015*. 18 Maret 2018 <https://tekno.kompas.com/read/2017/04/27/10230077/pengguna.twitter.naik.9.juta.penambahan.tertinggi.sejak.2015>.
- Ipmawati J., Kusriani, Luthfi E. T., (2017). "Komparasi Teknik Klasifikasi Teks Mining Pada Analisis Sentimen". *Indonesian Journal on Networking and Security*. 6 (1), 29-30.
- Kamagi David H. & Hansun S. 2014, "Implementasi Data Mining dengan Algoritma C4.5 untuk Memprediksi Tingkat Kelulusan Mahasiswa", *Ultimatics*, Vol.6, No.1, dalam <http://library.umn.ac.id/jurnal/public/uploads/papers/pdf/be20fbb34fbc7c5d2fe06044edf6d37e.pdf>, diunduh pada 02 oktober 2017.
- Librian, Andy, Kuku, R. 2014. *Library Sastrawi Stemmer* <https://github.com/jsastrawi/jsastrawi>, di akses 12 Desember 2017.

- Liu, B., 2012. *Sentiment Analysis and Opinion Mining*. California: Morgan & Claypool
- Mandala, R. Bahan Kuliah Sistem Temu Balik Informasi. Institut Teknologi Bandung. Departemen Teknik Informatika. 2004
- Maulana, Aqmal., 2016. Twitter Rahasiakan Jumlah Pengguna di Indonesia. 13 Maret 2017. <http://www.cnnindonesia.com/teknologi/20160322085045-185-118939/twitter-rahasiakan-jumlah-pengguna-di-indonesia/>
- Muthia D.A. (2016). "Opinion Mining Pada Review Buku Menggunakan Algoritma Naïve Bayes". Jurnal Teknik Komputer Amik BSI. 2 (1), 1-7.
- M. Ramaswami, R. Bhaskaran, "A Study on Feature Selection Techniques in Educational Data Mining", Journal Of Computing, vol. 1, Issue 1, pp. 7-11. December 2009.
- NCBI, 2017. Table StopWord List. 13 Maret 2017 <https://www.ncbi.nlm.nih.gov/books/NBK3827/table/pubmedhelp.T.stopwords/>
- Nicol, J., 2013. *TweetPHP: Display tweets on your website using PHP*. 13 Maret 2017. <http://jonathannicol.com/blog/2013/06/20/tweetphp-display-tweets-on-your-website-using-php/>
- Nurina Sari B. "Implementasi Teknik Seleksi Fitur Information Gain pada Algoritma Klasifikasi Machine Learning", Jurnal STMIK AMIKOM Yogyakarta 7-11. December 2009, vol 5 (1), 1-2.
- Pang, B. & Lee, L., 2008. Opinion Mining and Sentiment Analysis. *Foundation and Trend in Information Retrieval Volume 2 issue 1-2*. pp.1-135.
- Patil, G., Galande, V., Kekan, V. & Dange, K., 2014. *Sentiment Analysis Using Support Machine. International Journal of Innovative Research in Computer and Communication Engineering*. pp. 2607-2612
- Pamungkas D.S., Setiyanto N.A., Dolphina E. (2015). "Analisis Sentiment pada Sosial Media Twitter Menggunakan Naïve Bayes Classifier Terhadap Kata Kunci "Kurikulum 2013". Jurnal Techno.COM. 14 (4), 299-314
- Saifinnuha, A.Z. (2013). "Penerapan Sentimen Anaalisis pada Twitter Berbahasa Indonesia untuk mendapatkan Rating Program Televisi Menggunakan Metode SVM (Support Vector Machine)". Skripsi S1 FILKOM UB Malang: tidak diterbitkan
- Sari, Betha Nurina, (2016). "Implementasi Teknik Seleksi Fitur Information Gain pada Algoritma Klasifikasi Machine Learning untuk Prediksi Performa Akademik Siswa". Jurnal STMIK AMIKOM Yogyakarta. 2 (9), 56-57
- Tala, F. Z., 2003. *A Study of Stemming Effects on Information Retrieval in Bahasa Indonesia*. Amsterdam: Institute for Logic, Language and Computation Universiteit van Amsterdam.

- Xhemali, D., J. Hinde, C. & G. Stone, R., 2009. Naïve Bayes vs. Decision Trees vs. Neural Networks in the Classification of Training Web Pages. *International Journal of Computer Science Issues*, 4(1), pp. 16-23.
- Wardana., 2010, Menjadi Master PHP dengan Framework CodeIgniter, ELEXMEDIA, Jakarta.
- Weiss, S. M., Indurkha, N. & Zhang, T., 2005. *Text Mining : Predictive methods for Analyzing Unstructured Information*. New York : Springer.
- Kamus Besar Bahasa Indonesia (KBBI) versi online, "Sentimen". 24 Oktober 2017 <https://kbbi.web.id/sentimen>

